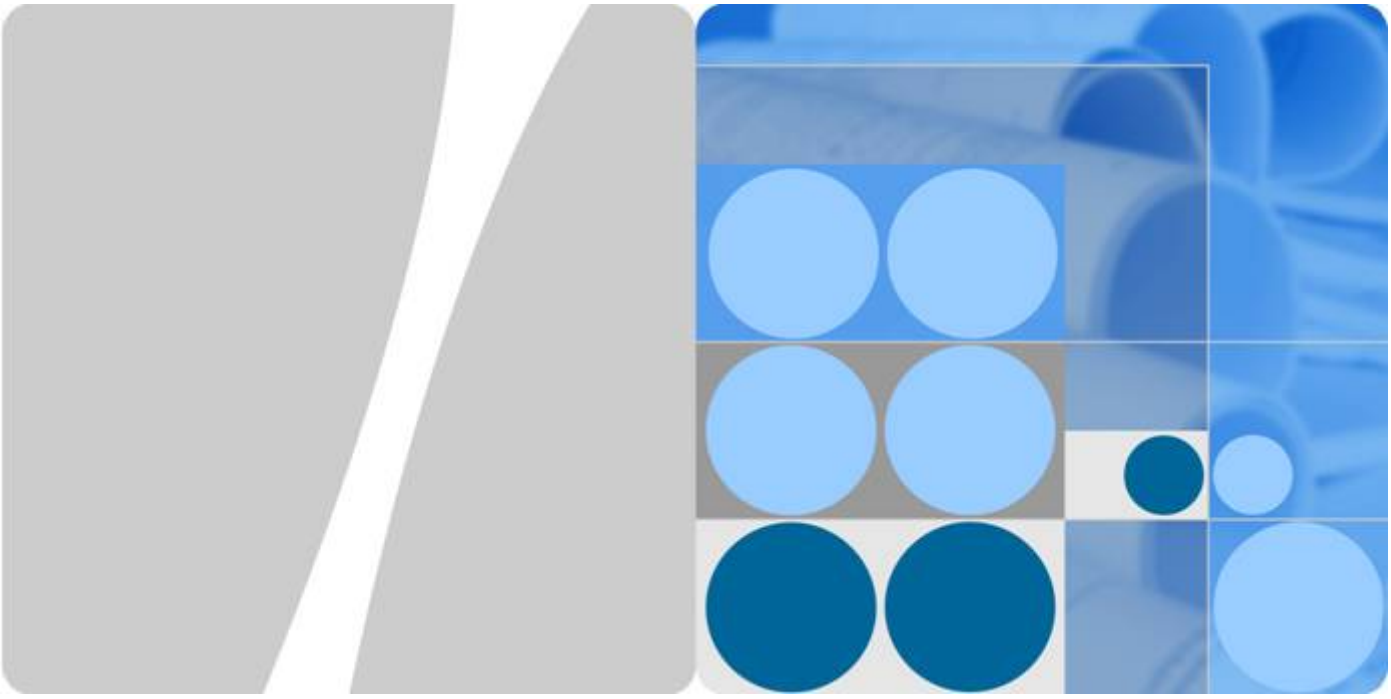


Data Lake Insight

User Guide

Date **2023-10-24**

Contents

1 Service Overview

1.1 What Is Data Lake Insight?

DLI Introduction

Data Lake Insight (DLI) is a serverless data processing and analysis service fully compatible with **Apache Spark** and **Apache Flink** ecosystems. It frees you from managing any servers.

DLI supports standard SQL and is compatible with Spark SQL and Flink SQL. It also supports multiple access modes, and is compatible with mainstream data formats. DLI supports SQL statements and Spark applications for heterogeneous data sources, including CloudTable, RDS, GaussDB(DWS), CSS, OBS, custom databases on ECSs, and offline databases.

Functions

You can query and analyze heterogeneous data sources such as RDS, and GaussDB(DWS) on the cloud using access methods, such as visualized interface, RESTful API, JDBC, and Beeline. The data format is compatible with five mainstream data formats: CSV, JSON, Parquet, and ORC.

- Basic functions
  - You can use standard SQL statements to query in SQL jobs.
  - Flink jobs support Flink SQL online analysis. Aggregation functions such as Window and Join, geographic functions, and CEP functions are supported. SQL is used to express service logic, facilitating service implementation.
  - For spark jobs, fully-managed Spark computing can be performed. You can submit computing tasks through interactive sessions or in batch to analyze data in the fully managed Spark queues.
- Federated analysis of heterogeneous data sources
  - Spark datasource connection: Data sources such as DWS, RDS, and CSS can be accessed through DLI.

  - Interconnection with multiple cloud services is supported in Flink jobs to form a rich stream ecosystem. The DLI stream ecosystem consists of cloud service ecosystems and open source ecosystems.
    - Cloud service ecosystem: DLI can interconnect with other services in Flink SQL. You can directly use SQL to read and write data from cloud services.
    - Open-source ecosystems: After connections to other VPCs are established through datasource connections, you can access all data sources and output targets (such as Kafka, HBase, and Elasticsearch) supported by Flink and Spark in your dedicated DLI queue.

- Storage-compute decoupling

  DLI is interconnected with OBS for data analysis. In this architecture where storage and compute are decoupled, resources of these two types are charged separately, helping you reduce costs and improving resource utilization.

  You can choose single-AZ or multi-AZ storage when you create an OBS bucket for storing redundant data on the DLI console. The differences between the two storage policies are as follows:

  - Multi-AZ storage means data is stored in multiple AZs, improving data reliability. If the multi-AZ storage is enabled for a bucket, data is stored in multiple AZs in the same region. If one AZ becomes unavailable, data can still be properly accessed from the other AZs. The multi-AZ storage is ideal for scenarios that demand high reliability. You are advised to use this policy.
  - Single-AZ storage means that data is stored in a single AZ, with lower costs.

DLI Core Engine: Spark+Flink

- Spark is a unified analysis engine that is ideal for large-scale data processing. It focuses on query, compute, and analysis. DLI optimizes performance and reconstructs services based on open-source Spark. It is compatible with the Apache Spark ecosystem and interfaces, and improves performance by 2.5x when compared with open-source Spark. In this way, DLI enables you to perform query and analysis of EB's of data within hours.
- Flink is a distributed compute engine that is ideal for batch processing, that is, for processing static data sets and historical data sets. You can also use it for stream processing, that is, processing real-time data streams and generating data results in real time. DLI enhances features and security based on the open-source Flink and provides the Stream SQL feature required for data processing.

Serverless Architecture

DLI is a serverless big data query and analysis service. It has the following advantages:

- Auto scaling: DLI ensures you always have enough capacity on hand to deal with any traffic spikes.

Accessing DLI

A web-based service management platform is provided. You can access DLI using the management console or HTTPS-based APIs, or connect to the DLI server through the JDBC client.

- Using the management console

  You can submit SQL, Spark, or Flink jobs on the DLI management console. Log in to the management console. Choose **EI Enterprise Intelligence** > **Data Lake Insight**.
- Using APIs

  If you need to integrate DLI into a third-party system for secondary development, you can call DLI APIs to use the service.

  For details, see *Data Lake Insight API Reference*.

1.2 Advantages

Full SQL Compatibility

You do not need a background in big data to use DLI for data analysis. You only need to know SQL, and you are good to go. The SQL syntax is fully compatible with the standard ANSI SQL 2003.

Decoupled Storage and Compute

DLI compute and storage loads are decoupled. This architecture allows you to flexibly configure storage and compute resources on demand, improving resource utilization and reducing costs.

Serverless DLI

DLI is fully compatible with **Apache Spark** and **Apache Flink** ecosystems and APIs. It is a serverless big data computing and analysis service that integrates real-time, offline, and interactive analysis. Offline applications can be seamlessly migrated to the cloud, reducing the migration workload. DLI provides a highly-scalable framework integrating batch and stream processing, allowing you to handle data analysis requests with ease. With a deeply optimized kernel and architecture, DLI delivers 100-fold performance improvement compared with the MapReduce model. Your analysis is backed by an industry-vetted 99.95% SLA.

Cross-Source Analysis

Analyze your data across databases. No migration required. A unified view of your data gives you a comprehensive understanding of your data and helps you innovate faster. There are no restrictions on data formats, cloud data sources, or whether the database is created online or off.

1.3 Application Scenarios

DLI is applicable to large-scale log analysis, federated analysis of heterogeneous data sources, and big data ETL processing.

Large-scale Log Analysis

- Gaming operations data analysis

  Different departments of a game company analyze daily new logs via the game data analysis platform to obtain required metrics and make decision based on the obtained metric data. For example, the operation department obtains required metric data, such as new players, active players, retention rate, churn rate, and payment rate, to learn the current game status and determine follow-up actions. The placement department obtains the channel sources of new players and active players to determine the platforms for placement in the next cycle.

- Advantages
  - Efficient Spark programming model: DLI directly ingests data from DIS and performs preprocessing such as data cleaning. You only need to edit the processing logic, without paying attention to the multi-thread model.
  - Ease of use: You can use standard SQL statements to compile metric analysis logic without paying attention to the complex distributed computing platform.

Federated Analysis of Heterogeneous Data Sources

- Digital service transformation for car companies

  In the face of new competition pressures and changes in travel services, car companies build the IoV cloud platform and IVI OS to streamline Internet applications and vehicle use scenarios, completing digital service transformation for car companies. This delivers better travel experience for vehicle owners, increases the competitiveness of car companies, and promotes sales growth. For example, DLI can be used to collect and analyze daily vehicle metric data (such as batteries, engines, tire pressure, and airbags), and give maintenance suggestions to vehicle owners in time.

- Advantages
  - No need for migration in multi-source data analysis: RDS stores the basic information about vehicles and vehicle owners, table store saves real-time vehicle location and health status, and DWS stores periodic metric statistics. DLI allows federated analysis on data from multiple sources without data migration.
  - Tiered data storage: Car companies need to retain all historical data to support auditing and other services that require infrequent data access. Warm and cold data is stored in OBS and frequently accessed data is stored in DWS, reducing the overall storage cost.
  - Rapid and agile alarm triggering: There are no special requirements for the CPU, memory, hard disk space, and bandwidth.

Big Data ETL Processing

- Carrier big data analysis

  Carriers typically require petabytes, or even exabytes of data storage, for both structured (base station details) and unstructured (messages and communications) data. They need to be able to access the data with extremely low data latency. It is a major challenge to extract value from this data efficiently. DLI provides multi-mode engines such as batch processing

and stream processing to break down data silos and perform unified data analysis.

- Advantages
  - Big data ETL: You can enjoy TB to EB-level data governance capabilities to quickly perform ETL processing on massive carrier data. Distributed datasets are provided for batch processing.
  - High Throughput, Low Latency: DLI uses the Dataflow model of Apache Flink, a real-time computing framework. High-performance computing resources are provided to consume data from your created Kafka, DMS Kafka, and MRS Kafka clusters. A single CU processes 1,000 to 20,000 messages per second.

1.4 Constraints

Constraints on Jobs

- Only the latest 100 jobs are displayed on DLI's SparkUI.
- A maximum of 1,000 job results can be displayed on the console. To view more or all jobs, export the job data to OBS.
- To export job run logs, you must have the permission to access OBS buckets. You need to configure a DLI job bucket on the **Global Configuration** > **Project** page in advance.
- **View Log** and **Export Log** buttons are not available for synchronization jobs and jobs running on the default queue.
- Only Spark jobs support custom images.
- An elastic resource pool supports a maximum of 32,000 CUs.

For details about job constraints, see **Job Management**.

Constraints on Queues

- A queue named **default** is preset in DLI for you to experience. Resources are allocated on demand.
- Queue types:
  - For SQL: Spark SQL jobs can be submitted to SQL queues.
  - For general purpose: The queue is used to run Spark programs, Flink SQL jobs, and Flink Jar jobs.

  The queue type cannot be changed. If you want to use another queue type, purchase a new queue.
- The region of a queue cannot be changed.
- A newly created queue can be scaled in or out only after a job is executed on the queue.
- DLI queues cannot access the Internet.

For more constraints on using a DLI queue, see **Queue Overview**.

Constraints on Resources

- **Database**
  - **default** is the database built in DLI. You cannot create a database named **default**.
  - DLI supports a maximum of 50 databases.
- **Table**
  - DLI supports a maximum of 5,000 tables.
  - DLI supports the following table types:
    - **MANAGED**: Data is stored in a DLI table.
    - **EXTERNAL**: Data is stored in an OBS table.
    - **View**: A view can only be created using SQL statements.
    - Datasource table: The table type is also **EXTERNAL**.
  - You cannot specify a storage path when creating a DLI table.
- **Data import**
  - Only OBS data can be imported to DLI or OBS.
  - You can import data in CSV, Parquet, ORC, JSON, or Avro format from OBS to tables created on DLI.
  - To import data in CSV format to a partitioned table, place the partition column in the last column of the data source.
  - The encoding format of imported data can only be UTF-8.
- **Data export**
  - Data in DLI tables (whose table type is **MANAGED**) can only be exported to OBS buckets, and the export path must contain a folder.
  - The exported file is in JSON format, and the text format can only be UTF-8.
  - Data can be exported across accounts. That is, after account B authorizes account A, account A has the permission to read the metadata and permission information of account B's OBS bucket as well as the read and write permissions on the path. Account A can export data to the OBS path of account B.
- **Package**
  - A package can be deleted, but a package group cannot be deleted.
  - The following types of packages can be uploaded:
    - **JAR**: JAR file
    - **PyFile**: User Python file
    - **File**: User file
    - **ModelFile**: User AI model file

For details about constraints on resources, see **Data Management**.

Constraints on Enhanced Datasource Connections

- Datasource connections cannot be created for the **default** queue.
- Flink jobs can directly access DIS, OBS, and SMN data sources without using datasource connections.
- **VPC Administrator** permissions are required for enhanced connections to use VPCs, subnets, routes, VPC peering connections.
- If you use an enhanced datasource connection, the CIDR block of the elastic resource pool or queue cannot overlap with that of the data source.
- Only queues bound with datasource connections can access datasource tables.
- Datasource tables do not support the preview function.
- When checking the connectivity of datasource connections, the constraints on IP addresses are as follows:
  - The IP address must be valid, which consists of four decimal numbers separated by periods (.). The value ranges from 0 to 255.
  - During the test, you can add a port after the IP address and separate them with colons (:). The port can contain a maximum of five digits. The value ranges from 0 to 65535.

    For example, **192.168.***xx.xx* or **192.168.***xx.xx***:8181**.
- When checking the connectivity of datasource connections, the constraints on domain names are as follows:
  - The domain name can contain 1 to 255 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
  - The top-level domain name must contain at least two letters, for example, **.com**, **.net**, and **.cn**.
  - During the test, you can add a port after the domain name and separate them with colons (:). The port can contain a maximum of five digits. The value ranges from 0 to 65535.

    For example, **example.com:8080**.

For more constraints on enhanced datasource connections, see **Enhanced Datasource Connection Overview**.

Constraints on Datasource Authentication

- Compared with datasource authentication provided by DLI, you are advised to use Data Encryption Worksop (DEW) to store data source authentication information.
- Only Spark SQL and Flink OpenSource SQL 1.12 jobs support datasource authentication.
- The version of the cluster where the queue belongs may not support datasource authentication. You are advised to create a queue to run Flink jobs.
- DLI supports four types of datasource authentication. Select an authentication type specific to each data source.
  - CSS: applies to 6.5.4 or later CSS clusters with the security mode enabled.
  - Kerberos: applies to MRS security clusters with Kerberos authentication enabled.

  - Kafka_SSL: applies to Kafka with SSL enabled.
  - Password: applies to GaussDB(DWS), RDS, DDS, and DCS.

For more constraints on datasource authentication, see **Datasource Authentication Introduction**.

Constraints on SQL Syntax

- Constraints on the SQL syntax:
  - You are not allowed to specify a storage path when creating a DLI table using SQL statements.
- Constraints on the size of SQL statements:
  - Each SQL statement should contain less than 500,000 characters.
  - The size of each SQL statement must be less than 1 MB.

Other Constraints

- For details about quota constraints, see **Quotas**.
- Recommended browsers for logging in to DLI:
  - Google Chrome 43.0 or later
  - Mozilla Firefox 38.0 or later
  - Internet Explorer 9.0 or later

1.5 Permissions Management

If you need to assign different permissions to employees in your enterprise to access your DLI resources, IAM is a good choice for fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, helping you securely access to your cloud resources.

With IAM, you can use your account to create IAM users for your employees, and assign permissions to the users to control their access to specific resource types. For example, some software developers in your enterprise need to use DLI resources but must not delete them or perform any high-risk operations. To achieve this result, you can create IAM users for the software developers and grant them only the permissions required for using DLI resources.

If your account does not require individual IAM users for permissions management, you may skip over this section.

DLI Permissions

By default, new IAM users do not have permissions assigned. You need to add the users to one or more groups, and attach permissions policies or roles to these groups. The users then inherit permissions from the groups to which they are added. After authorization, the users can perform specified operations on DLI based on the permissions.

DLI is a project-level service deployed and accessed in specific physical regions. To assign ServiceStage permissions to a user group, specify the scope as region-specific projects and select projects for the permissions to take effect. If **All**

projects is selected, the permissions will take effect for the user group in all region-specific projects. When accessing DLI, the users need to switch to a region where they have been authorized to use cloud services.

Type: There are roles and policies.

- Roles: A type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. Only a limited number of service-level roles are available. If one role has a dependency role required for accessing SA, assign both roles to the users. However, roles are not an ideal choice for fine-grained authorization and secure access control.
- Policies: A type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization, meeting requirements for secure access control. For example, you can grant DLI users only the permissions for managing a certain type of ECSs. For the actions supported by DLI APIs, see "Permissions Policies and Supported Actions" in the *Data Lake Insight API Reference*.

Table 1-1 DLI system permissions

Role/Policy Name	Description	Category
DLI FullAccess	Full permissions for DLI.	System-defined policy
DLI ReadOnlyAccess	Read-only permissions for DLI. With read-only permissions, you can use DLI resources and perform operations that do not require fine-grained permissions. For example, create global variables, create packages and package groups, submit jobs to the default queue, create tables in the default database, create datasource connections, and delete datasource connections.	System-defined policy
Tenant Administrator	Tenant administrator • Job execution permissions for DLI resources. After a database or a queue is created, the user can use the ACL to assign rights to other users. • Scope: project-level service	System-defined role

Role/Policy Name	Description	Category
DLI Service Admin	DLI administrator. • Job execution permissions for DLI resources. After a database or a queue is created, the user can use the ACL to assign rights to other users. • Scope: project-level service	System-defined role

Table 1-2 lists the common operations supported by each system policy. You can choose required system policies according to this table.

Table 1-2 Common operations supported by each system permission

Resource	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
Queue	DROP_QUEUE	Deleting a Queue	√	×	√	√
	SUBMIT_JOB	Submitting a job	√	×	√	√
	CANCEL_JOB	Terminating a Job	√	×	√	√
	RESTART	Restarting a queue	√	×	√	√
	GRANT_PRIVILEGE	Granting permissions to a queue	√	×	√	√
	REVOKE_PRIVILEGE	Revoking permissions to a queue	√	×	√	√
	SHOW_PRIVILEGES	Viewing the queue permissions of other users	√	×	√	√
Database	DROP_DATABASE	Deleting a database	√	×	√	√
	CREATE_TABLE	Creating a table	√	×	√	√
	CREATE_VIEW	Creating a view	√	×	√	√

Resource	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
	EXPLAIN	Explaining the SQL statement as an execution plan	√	×	√	√
	CREATE_ROLE	Creating a role	√	×	√	√
	DROP_ROLE	Deleting a role	√	×	√	√
	SHOW_ROLES	Displaying a role	√	×	√	√
	GRANT_ROLE	Binding a role	√	×	√	√
	REVOKE_ROLE	Unbinding a role	√	×	√	√
	SHOW_USERS	Displaying the binding relationships between all roles and users	√	×	√	√
	GRANT_PRIVILEGE	Granting permissions to the database	√	×	√	√
	REVOKE_PRIVILEGE	Revoking permissions to the database	√	×	√	√
	SHOW_PRIVILEGES	Viewing database permissions of other users	√	×	√	√
	DISPLAY_ALL_TABLES	Displaying tables in a database	√	√	√	√
	DISPLAY_DATABASE	Displaying databases	√	√	√	√

Resource	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
	CREATE_FUNCTION	Creating a function	√	×	√	√
	DROP_FUNCTION	Deleting a function	√	×	√	√
	SHOW_FUNCTIONS	Displaying all functions	√	×	√	√
	DESCRIBE_FUNCTION	Displaying function details	√	×	√	√
Table	DROP_TABLE	Deleting tables	√	×	√	√
	SELECT	Querying tables	√	×	√	√
	INSERT_INTO_TABLE	Inserting table data	√	×	√	√
	ALTER_TABLE_ADD_COLUMNS	Adding a column	√	×	√	√
	INSERT_OVERWRITE_TABLE	Overwriting a table	√	×	√	√
	ALTER_TABLE_RENAME	Renaming a table	√	×	√	√
	ALTER_TABLE_ADD_PARTITION	Adding partitions to the partition table	√	×	√	√
	ALTER_TABLE_RENAME_PARTITION	Renaming a table partition	√	×	√	√
	ALTER_TABLE_DROP_PARTITION	Deleting partitions from a partition table	√	×	√	√
	SHOW_PARTITIONS	Displaying all partitions	√	×	√	√

Resource	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
	ALTER_TABLE_RECOVER_PARTITION	Restoring table partitions	√	×	√	√
	ALTER_TABLE_SET_LOCATION	Setting the partition path	√	×	√	√
	GRANT_PRIVILEGE	Granting permissions to the table	√	×	√	√
	REVOKE_PRIVILEGE	Revoking permissions to the table	√	×	√	√
	SHOW_PRIVILEGES	Viewing table permissions of other users	√	×	√	√
	DISPLAY_TABLE	Displaying a table	√	√	√	√
	DESCRIBE_TABLE	Displaying table information	√	×	√	√

1.6 Quotas

What Is a Quota?

A quota limits the quantity of a resource available to users, thereby preventing spikes in the usage of the resource.

You can also request for an increased quota if your existing quota cannot meet your service requirements.

How Do I View My Quotas?

1. Log in to the management console.
2. Click  in the upper left corner and select a region and a project.
3. Click the **My Quota** icon  in the upper right corner of the page.
   The **Service Quota** page is displayed.
4. View the used and total quota of each type of resources on the displayed page.

If a quota cannot meet service requirements, increase a quota.

How Do I Apply for a Higher Quota?

The system does not support online quota adjustment. To increase a resource quota, dial the hotline or send an email to the customer service. We will process your application and inform you of the progress by phone call or email.

Before you contact customer service, prepare the following information:

- Account name, project name, and project ID

  Log in to the management console, click the username in the upper-right corner, choose **My Credentials**, and obtain the domain name, project name, and project ID.
- Quota information, including:
  - ServiceName
  - Quota type
  - Required quota

Learn how to obtain the service hotline and email address.

1.7 Basic Concepts

Tenant

DLI allows multiple organizations, departments, or applications to share resources. A logical entity, also called a tenant, is provided to use diverse resources and services. A mode involving different tenants is called multi-tenant mode. A tenant corresponds to a company. Multiple sub-users can be created under a tenant and are assigned different permissions.

Project

A project is a collection of resources accessible to services. In a region, an account can create multiple projects and assign different permissions to different projects. Resources used for different projects are isolated from one another. A project can either be a department or a project team.

Database

A database is a warehouse where data is organized, stored, and managed based on the data structure. DLI management permissions are granted on a per database basis.

In DLI, tables and databases are metadata containers that define underlying data. The metadata in the table shows the location of the data and specifies the data structure, such as the column name, data type, and table name. A database is a collection of tables.

Metadata

Metadata is used to define data types. It describes information about the data, including the source, size, format, and other data features. In database fields, metadata interprets data content in the data warehouse.

Compute Resource

Queues in DLI are computing resources, which are the basis for using DLI. SQL jobs and Spark jobs performed by users require computing resources.

Storage Resource

Storage resources in DLI are used to store data of databases and DLI tables. To import data to DLI, storage resources must be prepared. The storage resources reflect the volume of data you are allowed to store in DLI.

SQL Job

SQL job refers to the SQL statement executed in the SQL job editor. It serves as the execution entity used for performing operations, such as importing and exporting data, in the SQL job editor.

Spark Job

Spark jobs are those submitted by users through visualized interfaces and RESTful APIs. Full-stack Spark jobs are allowed, such as Spark Core, DataSet, MLlib, and GraphX jobs.

OBS Table, DLI Table, and CloudTable Table

The table type indicates the storage location of data.

- OBS table indicates that data is stored in the OBS bucket.
- DLI table indicates that data is stored in the internal table of DLI.
- CloudTable table indicates that data is stored in CloudTable.

You can create a table on DLI and associate the table with other services to achieve querying data from multiple data sources.

Constants and Variables

The differences between constants and variables are as follows:

- During the running of a program, the value of a constant cannot be changed.
- Variables are readable and writable, whereas constants are read-only. A variable is a memory address that contains a segment of data that can be changed during program running. For example, in **int a = 123**, **a** is an integer variable.

2 Getting Started

2.1 Creating and Submitting a Spark SQL Job

Scenario

DLI can query data stored in OBS. This section describes how to us a Spark SQL job on DLI to query OBS data.

Procedure

You can use DLI to submit a Spark SQL job to query data. The general procedure is as follows:

Step 1: Upload Data to OBS

Step 2: Create a Queue

Step 3: Create a Database

Step 4: Create a Table

Step 5: Query Data

Step 1: Upload Data to OBS

Before you use DLI to query and analyze data, upload data files to OBS.

1. Go to the DLI console.
2. In the service list, click **Object Storage Service** under **Storage**. The OBS console page is displayed.
3. Create a bucket. In this example, the bucket name is **obs1**.
   a. Click **Create Bucket** in the upper right corner.
   b. On the displayed **Create Bucket** page, enter the **Bucket Name**. Retain the default values for other parameters or adjust them as needed.

      NOTE

      You must select the same region as the DLI management console.

c. Click **Create Now**.

4. Click **obs1** to access its **Objects** tab page.
5. Click **Upload Object**. In the displayed dialog box, drag a desired file or folder, for example, **sampledata.csv** to the **Upload Object** area. Then, click **Upload**.

   You can create a **sampledata.txt** file, copy the following content separated by commas (,), and save the file as **sampledata.csv**.

   ```
   12,test
   ```

   After the file is uploaded successfully, the file path is **obs://obs1/sampledata.csv**.

   NOTE

   - For more information about OBS operations, see the *Object Storage Service Console Operation Guide*.
   - For more information about the tool, see the *OBS Tool Guide*.
   - You are advised to use an OBS tool, such as OBS Browser+, to upload large files because OBS Console has restrictions on the file size and quantity.
     - OBS Browser+ is a graphical tool that provides complete functions for managing your buckets and objects in OBS.

Step 2: Create a Queue

A queue is the basis for using DLI. Before executing an SQL job, you need to create a queue.

- An available queue **default** is preset in DLI.
- You can also create queues as needed.

  a. Log in to the DLI management console.

  b. In the left navigation pane of the DLI management console, choose **SQL Editor**.

  c. On the left pane, select the **Queues** tab, and click  next to **Queues**.

  For details, see Creating a Queue.

Step 3: Create a Database

Before querying data, create a database, for example, **db1**.

NOTE

The **default** database is a built-in database. You cannot create the **default**. database.

1. In the left navigation pane of the DLI management console, choose **SQL Editor**.
2. In the editing window on the right of the **SQL Editor** page, enter the following SQL statement and click **Execute**. Read and agree to the privacy agreement, and click **OK**.

   ```
   create database db1;
   ```

   After the database is successfully created, click  in the middle pane to refresh the database list. The new database **db1** is displayed in the list.

NOTE

When you execute a query on the DLI management console for the first time, you need to read the privacy agreement. You can perform operations only after you agree to the agreement. For later queries, you will not need to read the privacy agreement again.

Step 4: Create a Table

After database **db1** is created, create a table (for example, **table1**) containing data in the sample file **obs://obs1/sampledata.csv** stored on OBS in **db1**.

1. In the SQL editing window of the **SQL Editor** page, select the **default** queue and database **db1**.
2. Enter the following SQL statement in the job editor window and click **Execute**:
   ```
   create table table1 (id int, name string) using csv options (path 'obs://obs1/sampledata.csv');
   ```
   After the table is successfully created, click the **Databases** tab then **db1**. The created table **table1** is displayed in the table list.

Step 5: Query Data

After performing the preceding steps, you can start querying data.

1. In the **Table** tab on the **SQL Editor** page, double-click the created table **table1**. The SQL statement is automatically displayed in the SQL job editing window in the right pane. Run following statement to query 1,000 records in the **table1** table:
   ```
   select * from db1.table1 limit 1000;
   ```
2. Click **Execute**. The system starts the query.

   After the SQL statement is successfully executed or fails to be executed, you can view the query result on the **View Result** tab under the SQL job editing window.

2.2 Developing and Submitting a Spark SQL Job Using the TPC-H Sample Template

DLI allows you to customize query templates or save frequently used SQL statements as templates to facilitate SQL operations. After templates are saved, you do not need to write SQL statements. You can directly perform the SQL operations using the templates.

The current system provides various standard TPC-H query statement templates. You can select a template as needed. This example shows how to use a TPC-H template to develop and submit a Spark SQL job.

Step 1: Log In to the Management Console

Step 2: Execute the TPC-H Sample Template and View the Result

For details about the templates, see SQL Template Management.

Step 1: Log In to the Management Console

1. Log in to the management console.
2. Click **Service List** and choose **Analytics** > **Data Lake Insight**.

   NOTE

   You need to perform authorization when accessing the DLI management console for the first time. For details, see "Global Configuration" > "Service Authorization" in *Data Lake Insight User Guide*.

Step 2: Execute the TPC-H Sample Template and View the Result

1. On the DLI management console, choose **Job Templates** > **SQL Templates**, and click the **Sample Templates** tab. Locate the **Q1_Price_summary_report_query** template under **tpchQuery**, and click **Execute** in the **Operation** column. The **SQL Editor** page is displayed.
2. In the upper part of the editing window, set **Engine** to **spark**, **Queues** to **default**, and **Databases** to **default**, and click **Execute**.
3. View the query result in the **View Result** tab in the lower part of the SQL Editor page.

This example uses the **default** queue and database preset in the system as an example. You can also run query statements on a self-created queue and database.

For details about how to create a queue, see "Creating a Queue" in *Data Lake Insight User Guide*. For details about how to create a database, see Creating a Database.

2.3 Creating and Submitting a Spark Jar Job

Scenario

DLI can query data stored in OBS. This section describes how to use a Spark Jar job on DLI to query OBS data in real time.

Procedure

You can use DLI to submit Spark jobs for real-time computing. The general procedure is as follows:

Step 1: Upload Data to OBS

Step 2: Create a Queue

Step 3: Create a Package

Step 4: Submit a Spark Job

Step 1: Upload Data to OBS

Write a Spark Jar job program, and compile and pack it as **spark-examples.jar**. Perform the following steps to submit the job:

Before submitting Spark Jar jobs, upload data files to OBS.

1. Log in to the DLI console.
2. In the service list, click **Object Storage Service** under **Storage**. The OBS console page is displayed.
3. Create a bucket. In this example, name it **dli-test-obs01**.
   a. Click **Create Bucket**.
   b. On the displayed **Create Bucket** page, enter the **Bucket Name**. Retain the default values for other parameters or set them as required.

      NOTE

      When creating an OBS bucket, you must select the same region as the DLI management console.

   c. Click **Create Now**.
4. Click **dli-test-obs01** to switch to the **Objects** tab page.
5. Click **Upload Object**. In the dialog box displayed, drag or add files or folders, for example, **spark-examples.jar**, to the upload area. Then, click **Upload**.

   After the file is uploaded successfully, the file path is **obs://dli-test-obs01/spark-examples.jar**.

   NOTE

   - For more information about OBS operations, see the *Object Storage Service Console Operation Guide*.
   - For more information about the tool, see the *OBS Tool Guide*.
   - You are advised to use an OBS tool, such as OBS Browser+, to upload large files because OBS Console has restrictions on the file size and quantity.
     - OBS Browser+ is a graphical tool that provides complete functions for managing your buckets and objects in OBS. You are advised to use this tool to create buckets or upload objects.

Step 2: Create a Queue

If you submit a Spark job for the first time, you need to create a queue first. For example, create a queue named **sparktest** and set **Queue Type** to **General Queue**.

1. Log in to the DLI management console.
2. In the navigation pane of the DLI management console, choose **Resources** > **Queue Management**.
3. In the upper right corner of the **Queue Management** page, click **Create Queue** to create a queue.
4. Create a queue, name it **sparktest**, and set the queue usage to for general purpose. For details, see Creating a Queue.
5. Click **Create Now** to create a queue.

Step 3: Create a Package

Before submitting a Spark job, you need to create a package, for example, **spark-examples.jar**.

1. In the navigation pane on the left of the DLI console, choose **Data Management** > **Package Management**.
2. On the **Package Management** page, click **Create** in the upper right corner to create a package.
3. In the **Create Package** dialog box, set **Type** to **JAR**, **OBS Path** to the path of the spark-examples.jar package in **Step 1: Upload Data to OBS**, and **Group** to **Do not use**.
4. Click **OK**.

   You can view and select the package on the **Package Management** page.

For details about how to create a package, see "Creating a Package".

Step 4: Submit a Spark Job

1. On the DLI management console, choose **Job Management** > **Spark Jobs** in the navigation pane on the left. On the displayed page, click **Create Job**.
2. On the Spark job editing page, set **Queues** to the queue created in **Step 2: Create a Queue** and **Application** to the package created in **Step 3: Create a Package**.

   For details about other parameters, see the description of the Spark job editing page in "Creating a Spark Job".
3. Click **Execute** in the upper right corner of the Spark job editing window, read and agree to the privacy agreement, and click **OK**. Submit the job. A message is displayed, indicating that the job is submitted.
4. (Optional) Switch to the **Job Management** > **Spark Jobs** page to view the status and logs of the submitted Spark job.

   NOTE

   When you click **Execute** on the DLI management console for the first time, you need to read the privacy agreement. Once agreed to the agreement, you will not receive any privacy agreement messages for subsequent operations.

2.4 Creating and Submitting a Flink SQL Job

Scenario

DLI Flink jobs can use other cloud services as data sources and sink streams for real-time compute. This section describes how to create and submit a Flink SQL job that uses DIS as the input stream and DMS Kafka as the output stream.

Procedure

You need to create a Flink SQL job that has an input stream and an output stream. The input stream is used to read data from DIS, and the output stream is used to write data to Kafka. The procedure is as follows:

Step 1: Prepare Data Sources and Data Output Channels

Step 2: Create an OBS Bucket for Saving Outputs

Step 3: Create a Queue

Step 1: Prepare Data Sources and Data Output Channels

In this example, DIS is the input stream. Distributed Message Service (DMS) Kafka is the output stream.

For more information about Flink job data, see **Preparing Flink Job Data**.

Enable DIS to import data into DLI. For details, see **Creating a DIS Stream** in the Data Ingestion Service User Guide.

- Create a DIS stream for the job input stream.
  a. Log in to the DIS console.
  b. In the upper left corner of the management console, select the target region and project.
  c. On the **Overview** page, click **Buy Stream** and configure stream parameters. The channel information is as follows:
     - **Region**: Select the region where DLI is located.
     - **Stream Name**: **csinput**
     - **Stream Type**: **Common**
     - **Partitions**: **1**
     - **Data Retention (hours)**: **24**
     - **Source Data Type**: **BLOB**
     - **Auto Scaling**: **disabled**
     - **Enterprise project**: **default**
     - **Advanced Settings**: Skip it.
  d. Click **Buy Now**. The **Details** page is displayed.
  e. Click **Submit**.
- Create a Kafka platinum instance for the job output stream.

  For details, see **Creating a Queue** in Distributed Message Service Kafka User Guide.

  a. Before creating a Kafka instance, ensure the availability of resources, including a virtual private cloud (VPC), subnet, security group, and security group rules.

     The created VPC and the Kafka instance you will create must be in the same region. For more information, see **Managing Kafka Premium Instances** > **Preparing the Environment** in the *Distributed Message Service User Guide*.

b. Log in to the DMS console.
c. Select a region in the upper left corner.
d. Choose **DMS for Kafka** form the navigation pane on the left, click **Buy Instance** in the upper right corner, and set related parameters. The instance information is as follows:
   - **Region**: Select the region where DLI is located.
   - **Project**: Keep the default value.
   - **AZ**: Keep the default value.
   - **Instance Name**: **kafka-dliflink**
   - **Enterprise Project**: **default**
   - **Version**: Keep the default value.
   - **CPU Architecture**: Keep the default value.
   - **Specifications**: Select the specification as needed.
   - **Brokers**: Keep the default value.
   - **Storage Space**: Keep the default value.
   - **Capacity Threshold Policy**: Keep the default value.
   - **VPC** and **Subnet**: Select the VPC and subnet created in **a**.
   - **Security Group**: Select the security group created in **a**.
   - **Manager Username**: Enter **dliflink** (used to log in to the instance management page).
   - **Password**: **** (The system cannot detect your password.)
   - **Confirm Password**: ****
   - **Advanced Settings**: Enable **Kafka SASL_SSL** and configure the username and password for SSL authentication as prompted. You do not need to set other parameters.
e. Click **Buy**. The confirmation page is displayed.
f. Click **Submit**.
g. On the DMS for Kafka console, click **Kafka Premium** and click the name of the Kafka instance, for example, **kafka-dliflink**. The instance details page is displayed.
h. Locate the SSL certificate in **Basic Information** > **Advanced Settings**, and click **Download**. Download the package to the local PC and decompress it to obtain the client certificate file **client.truststore.jks**.

Step 2: Create an OBS Bucket for Saving Outputs

In this example, you need to enable OBS for job JobSample to provide DLI Flink jobs with the functions of checkpoint, saving job logs, and commissioning test data.

For details about how to create a bucket, see "Creating a Bucket" in the *Object Storage Service Console Operation Guide*.

1. In the navigation pane on the OBS management console, select **Object Storage**.
2. In the upper right corner of the page, click **Create Bucket** and set bucket parameters.
   - **Region**: Select the region where DLI is located.
   - **Bucket Name**: Enter a bucket name.
   - **Default Storage Class**: **Standard**
   - **Bucket Policy**: **Private**
   - **Default Encryption**: **Do not enable**
   - **Direct Reading**: **Do not enable**
   - **Enterprise project**: **default**
   - **Tags**: Leave it blank.
3. Click **Create Now**.

Step 3: Create a Queue

You cannot create a DLI Flink SQL job on the existing default queue. Instead, you need to create a queue, for example, a queue named **Flinktest**. For details, see Creating a Queue.

1. Go to the DLI console.
2. On the **Overview** page of the DLI management console, click **Buy Queue** in the upper right corner.
3. Configure the following parameters:
   - **Name**: **Flinktest**
   - **Queue Usage**: Select **For general purpose** and **Dedicated Resource Mode**.
   - **CU Specifications**: **16 CUs**
   - **Enterprise project**: **default**
   - **Description**: Leave it blank.
   - **Advanced Settings**: **Custom**
   - **CIDR Block**: The configured CIDR block cannot conflict with the Kafka subnet CIDR block.
4. Click **Buy** to confirm the configuration.
5. Confirm the configuration and submit the request.

Step 4: Create an Enhanced Datasource Connection

You need to create an enhanced datasource connection for the Flink job. For details, see "Creating an Enhanced Datasource Connection".

NOTE

- The CIDR block of the DLI queue bound with a datasource connection cannot overlap with the CIDR block of the data source.
- Datasource connections cannot be created for the **default** queue.
- To access a table across data sources, you need to use a queue bound to a datasource connection.

1. In the navigation pane of the DLI management console, choose **Datasource Connections**.
2. Click the **Enhanced** tab and click **Create**. Set the following parameters:
   - **Connection Name**: diskafka
   - **Bind Queue**: flinktest
   - **VPC**: vpc-dli
   - **Subnet**: dli-subnet

     NOTE

     The VPC and subnet must be the same as those of the Kafka instance.
3. Click **OK**.
4. On the **Enhanced** tab page, click the created connection **diskafka** to view its **VPC Peering ID** and **Connection Status**. If the status is **Active**, the connection is successful.

Step 5: Create a Datasource Authentication

For details about how to create data source authentication, see "Datasource Authentication".

1. Upload the Kafka authentication file **client.truststore.jks** obtained in **Step 1: Prepare Data Sources and Data Output Channels** to the OBS bucket **smoke-test** created in **Step 2: Create an OBS Bucket for Saving Outputs**.
2. On the DLI management console, click **Datasource Connections**.
3. On the **Datasource Authentication** tab page, click **Create** to add an authentication information. Set the following parameters:
   - **Authentication Certificate**: Flink
   - **Type**: Kafka_SSL
   - **Truststore Path**: **obs://smoke-test/client.truststore.jks**
   - **Truststore password**: dms@kafka

   You do not need to set other parameters.
4. Click **OK**.

Step 6: Configure Security Group Rules and Test Address Connectivity

1. On the DLI management console, click **Resources** > **Queue Management**, select the bound queue, and click the arrow next to the queue name to view the network segment information of the queue.
2. Log in to the DMS for Kafka console, click **Kafka Premium**, and click the name of the created Kafka instance, for example, **kafka-dliflink**. The instance information page is displayed.
3. On the **Basic Information** page, obtain the Kafka connection address and port number in the **Connection Address** area.
4. On the **Basic Information** page, click the security group name in the **Security Group** area.
5. On the security group configuration page of the Kafka instance, choose **Inbound Rules** > **Add Rule**. Set **Protocol** to **TCP**, **Port** to **9093**, and **Source** to the CIDR block of the DLI queue. Click **OK**.
6. Log in to the DLI management console and choose **Resources** > **Queue Management**. In the row of the Flink queue, choose **More** > **Test Address Connectivity**. In the **Address** text box, enter the Kafka connection address and port number in the format of IP address:port number, and click **Test**. The subsequent operations can be performed only when the address is reachable. Note that multiple addresses must be tested separately.

Step 7: Create a Flink SQL Job

After the data source and data output channel are prepared, you can create a Flink SQL job.

1. In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.
2. In the upper right corner of the **Flink Jobs** page, click **Create Job**. Set the following parameters:
   - **Type**: **Flink SQL**
   - **Name**: **DIS-Flink-Kafka**
   - **Description**: Leave it blank.
   - **Template Name**: Do not select any template.
3. Click **OK** to enter the editing page.
4. Edit the Flink SQL job

   Enter SQL statements in the editing window. The example statements are as follows. Note that the values of the parameters in bold must be changed according to the comments.

```
CREATE SOURCE STREAM car_info (
  a1 string,
  a2 string,
  a3 string,
  a4 INT
)
WITH (
  type = "dis",
  region = "xxx",// Region where the current DLI queue is located
  channel = "csinput",
  encode = "csv",
  FIELD_DELIMITER = ";"
```

```
);

 CREATE SINK STREAM kafka_sink (
  a1 string,
  a2 string,
  a3 string,
  a4 INT
) // Output field

WITH (
  type="kafka",
Change kafka_bootstrap_servers = "192.x.x.x:9093, 192.x.x.x:9093, 192.x.x.x:9093",// Connection address of the Kafka instance
kafka_topic = "testflink", // Topic to be written to Kafka. Log in to the Kafka console, click the name of the created Kafka instance, and view the topic name on the Topic Management page.
  encode = "csv", // Encoding format, which can be JSON or CSV.
  kafka_certificate_name = "Flink",// The value is the name of the Kafka datasource authentication created in Step 7.
  kafka_properties_delimiter = ",",
  // Replace xxx in username and password in kafka_properties with the username and password for creating SSL authentication for Kafka in step 2.
  kafka_properties = "sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule required username=\"xxx\" password=\"xxx\";,sasl.mechanism=PLAIN,security.protocol=SASL_SSL"
);

INSERT INTO kafka_sink
SELECT * FROM car_info;

CREATE sink STREAM car_info1 (
  a1 string,
  a2 string,
  a3 string,
  a4 INT
)
WITH (
  type = "dis",
  region = "xxx",// Region where the current DLI queue is located
  channel = "csinput",
  encode = "csv",
  FIELD_DELIMITER = ";"
);

insert into car_info1 select 'id','owner','brand',1;
insert into car_info1 select 'id','owner','brand',2;
insert into car_info1 select 'id','owner','brand',3;
insert into car_info1 select 'id','owner','brand',4;
insert into car_info1 select 'id','owner','brand',5;
insert into car_info1 select 'id','owner','brand',6;
insert into car_info1 select 'id','owner','brand',7;
insert into car_info1 select 'id','owner','brand',8;
insert into car_info1 select 'id','owner','brand',9;
insert into car_info1 select 'id','owner','brand',10;
```

5. Click **Check Semantics**.
6. Set job running parameters. The mandatory parameters are as follows:
   - **Queue**: Flinktest
   - **CUs**: 2
   - **Job Manager CUs**: 1
   - **Parallelism**: 1
   - **Save Job Log**: selected
   - **OBS Bucket**: Select the OBS bucket for storing job logs. You need the permissions to access this bucket.

   You do not need to set other parameters.

7. Click **Save**.
8. Click **Start**. On the displayed **Start Flink Job** page, confirm the job specifications and the price, and click **Start Now** to start the job.

   After the job is started, the system automatically switches to the **Flink Jobs** page, and the created job is displayed in the job list. You can view the job status in the **Status** column. After a job is successfully submitted, **Status** of the job will change from **Submitting** to **Running**.

   If **Status** of a job is **Submission failed** or **Running exception**, the job fails to be submitted or fails to run. In this case, you can move the cursor over the status icon to view the error details. You can click to copy these details. After handling the fault based on the provided information, resubmit the job.
9. After the job is complete, you can log in to the management console of Distributed Message Service for Kafka to view the corresponding Kafka premium instance. Click the instance name, click the **Message Query** tab, select the Kafka topic written in the Flink SQL job, and click **Search**. In the **Operation** column, click **View Message Body** to view the written message content.

3 DLI Console Overview

The **Overview** page of the DLI console provides you with the DLI workflow and resource usage.

How to Use DLI

The process of using DLI is as follows:

1. **Create a queue.**

   Queues are DLI's compute resources. There are SQL queues and general-purpose queues. For a SQL queue, you can only submit Spark SQL jobs. For a general-purpose queue, you can submit Spark programs, Flink OpenSource SQL jobs, and Flink Jar jobs.

2. **Prepare data.**

   Create databases and tables before you run a Spark SQL job. Upload a program package before you run a Spark job or a Flink Jar job.

3. **Edit and submit a job.**

   After you set the job parameters, you can submit the job.

4. **View job status.**

   Go to the **Job Management** page to view the job execution status.

Queue Usage (hours)

The overview page provides usage durations of all queues or a specific queue.

- Usage Usage (hours): an overview of the usage of all resources.
- Selected Queue Usage (hours): recent usage of a single queue.

4 SQL Editor

Introduction

You can edit and run SQL statements in the SQL job editor to execute data query.

The editor supports SQL:2003 and is compatible with Spark SQL. For details about the syntax, see .

To access the SQL editor, choose **SQL Editor** in the left navigation pane of the DLI console, or click **Create Job** in the upper right corner of the **Job Management** > **SQL Jobs** page.

This topic describes the main functions of the SQL editor.

Notes

- If you access the SQL editor for the first time, the system prompts you to set a bucket for DLI jobs. The created bucket is used to store temporary data generated by DLI, such as job logs.

  You cannot view job logs if you choose not to create the bucket. The bucket name will be set by the system.

  On the OBS console, you can configure lifecycle rules for a bucket to periodically delete objects in it or change object storage classes.
- SQL statements can be executed in batches on the SQL editor page.
- Commonly used keyworks in the job editing window are highlighted in different colors.
- Both single-line comment and multi-line comment are allowed. Use two consecutive hyphens (--) in each line to comment your statements.

Navigation pane

The navigation pane on the left consists of **Databases**, **Queues**, and **Templates** tabs.

Table 4-1 Navigation pane buttons

No.	Name	Description
1	Databases	Displays all the existing databases and tables in these databases. • Click a database name to view the tables in the database. • Click a table name to view the metadata in the table. A maximum of 20 metadata records can be displayed. • After you double-click a table name, a SQL query statement is automatically entered in the editing window.
2	Queues	Displays existing queues.
3	Templates	Click the drop-down button to view 22 built-in standard TPC-H query templates and custom templates.

SQL Editing Window

SQL job editing window is displayed in the upper right part of the page.

The SQL statement editing area is below the operation bar. For details about keyboard shortcuts, see **Table 4-3**.

Table 4-2 Components of the SQL job editing window

No.	Button & Drop-Down List	Description
2	Queues	Select a queue from the drop-down list box. If no queue is available, the **default** queue is displayed. Refer to **Creating a Queue** and create a queue. SQL jobs can be executed only on SQL queues.
3	Database	Select a database from the drop-down list box. If no database is available, the **default** database is displayed. For details about how to create a database, see **Creating a Database or a Table**. **NOTE** If you specify the database in the SQL statements, the database you choose from the drop-down list will not be used.
4	Execute	Click this button to run the SQL statements in the job editing window.
5	Format	Click this button to format the SQL statements.
6	Syntax Reference	Click this button to view the *Data Lake Insight SQL Syntax Reference*.

No.	Button & Drop-Down List	Description
7	Settings	Add parameters and tags. **Parameter Settings**: Set parameters in key/value format for SQL jobs. **Tags**: Set tags in key/value format for SQL jobs.
8	More	The drop-down list includes the following options: • Click **Verify Syntax** to check whether the SQL statements are correct. • Click **Set as Template** to set SQL statements as a template. For details, see **Managing SQL Templates**. • Click **Change Theme** to switch between dark and light modes.

Table 4-3 Keyboard shortcut description

Shortcut	Description
Ctrl+Enter	Execute SQL statements. You can run SQL statements by pressing **Ctrl+R** or **Ctrl + Enter** on the keyboard.
Ctrl+F	Search for SQL statements. You can press Ctrl+F to search for a required SQL statement.
Shift+Alt+F	Format SQL statements. You can press **Shift + Alt + F** to format a SQL statement.
Ctrl+Q	Syntax verification. You can press **Ctrl + Q** to verify the syntax of SQL statements.
F11	Full screen. You can press **F11** to display the SQL Job Editor window in full screen. Press **F11** again to leave the full screen.

Executed Queries (Last Day) and View Result

After the SQL job is executed, you can view the execution history and result in the lower part of the editing area.

- **Executed Queries (Last Day)**

  You can filter the execution history in the following ways:

  – In the search box in the upper right corner of the **Executed Queries (Last Day)** pane, select a queue name or enter an execution statement in the search box.

- In the list, click the icon next to **Created** and choose **Ascending** or **Descending**.
- Select a job status from the **Status** list.

Table 4-4 Area description

Area	Description
Executed Queries (Last Day)	The latest daily information about the submitted jobs, including the following items: • **Queues**: Queue name • **Username**: User who executes the SQL statements • **Type**: Type of the SQL job • **Status**: Execution status of the SQL job • **Query** • **Created** • **Operation** – **Edit**: Edit the SQL statement. – **SparkUI**: Switch to the SparkUI page to view the SQL statement execution process. **NOTE** When you execute a job on a created queue, the cluster is restarted. It takes about 10 minutes. If you click **SparkUI** before the cluster is created, an empty **projectID** will be cached. The SparkUI page cannot be displayed. You are advised to use a dedicated queue so that the cluster will not be released. Alternatively, wait for a while after the job is submitted (the cluster is created), and then check **SparkUI**. Currently, only the latest 100 job information records are displayed on the SparkUI of DLI. This function is not supported for synchronization jobs and jobs running on the default queue. – **More**: The following operations vary depending on the SQL job types and running status. **Cancel**: Cancel a SQL job that is running or being submitted. **Re-execute**: Execute the SQL statement again. **View Result**: View the execution result of a QUERY job. **Export Result**: Export the execution results of a QUERY job to a specified OBS path. **View Log**: View the OBS path for storing SQL statement execution logs. **Export Log**: Export SQL statement execution logs. **NOTE** To export the logs, you need to obtain the permission to create an OBS bucket. **View Log** and **Export Log** buttons are not available for synchronization jobs and jobs running on the default queue.

- **View Result**

Table 4-5 Operations in the result tab

Operation	Description
Clear the result	Clear the displayed SQL statement query results.
View chart/ table	Click  to view the query result in a chart or table.
Export the result	Click  to export the query result to OBS. For details, see **Exporting Query Results**. A maximum of 1000 records can be displayed in the query result on the console. To view more or all data, you can click **Export Result** to export the data to OBS.

SQL Query Procedure

1. Log in to the DLI management console. On the page displayed, choose **Job Management** > **SQL Jobs**. On the page displayed, click **Create Job**.

   NOTE

   On the SQL editor page, the system prompts you to create an OBS bucket to store temporary data generated by DLI jobs. In the **Set Job Bucket** dialog box, click **Setting**. On the page displayed, click the edit button in the upper right corner of the job bucket card. In the **Set Job Bucket** dialog box displayed, enter the job bucket path and click **OK**.

2. Select a queue from the queue list in the upper left corner of the SQL job editing window. For details about how to create a queue, see **Creating a Queue**.
3. In the upper right corner of the SQL job editing window, select a database, for example, **qw**, from the **Databases** drop-down list.
4. Create a table, for example, **qw**. For details about how to create a database and table, see **Creating a Database or a Table**.
5. In the SQL job editing window, enter the following SQL statement:

   ```
   SELECT * FROM qw.qw LIMIT 10;
   ```

   Alternatively, you can double-click the table name **qw**. The query statement is automatically entered in the SQL job editing window.
6. On top of the editing window, click **More** > **Verify Syntax** to check whether the SQL statement is correct.
   a. If the verification fails, check the SQL statement syntax by referring to *Data Lake Insight SQL Syntax Reference*.
   b. If the syntax verification is successful, click **Execute**. Read and agree to the privacy agreement. Click **OK** to execute the SQL statement.
   c. After the execution is complete, you can view the execution result in the area under the SQL job editing window.

7. (Optional) A maximum of 1000 records can be displayed in the query result on the current console. To view more or all data, click  to export the data to OBS.
8. (Optional) In the **View Result** tab, click  to display the query result in a chart. Click  to switch back to the table view.

   NOTE

   - If no column of the numeric type is displayed in the execution result, the result cannot be represented in charts.
   - You can view the data in a bar chart, line chart, or fan chart.
   - In the bar chart and line chart, the X axis can be any column, while the Y axis can only be columns of the numeric type. The fan chart displays the corresponding legends and indicators.

Quickly Importing SQL Statements

- Double-click a table name in the navigation pane on the left to import the query statement of the selected table into the SQL statement editing window, and then click **Execute** to query.
- You can click **More** and choose **Save as Template** to save the SQL statement as a template for future use.

  To use the SQL statement template, click **Templates** from the left pane of the SQL editor page. Double-click the required template in the template list, and modify it as required before executing the SQL statements.

5 Job Management

5.1 Overview

DLI Job Type

DLI provides the following job types:

- SQL job: SQL jobs provide you with standard SQL statements and are compatible with Spark SQL and Presto SQL (based on Presto). You can query and analyze heterogeneous data sources on the cloud through visualized APIs, JDBC, ODBC, or Beeline. SQL jobs are compatible with mainstream data formats such as CSV, JSON, Parquet, Carbon, and ORC.
- Flink job: Flink jobs are real-time streaming big data analysis service jobs running on the public cloud. In full hosting mode, you only need to focus on Stream SQL services and execute jobs instantly without being aware of compute clusters. Flink jobs are fully compatible with Apache Flink APIs.
- Spark job: Spark jobs provide fully-managed Spark compute services. You can submit jobs through the GUI or RESTful APIs. Full-stack Spark jobs, such as Spark Core, DataSet, Streaming, MLlib, and GraphX jobs, are supported.

Constraints

- Only the latest 100 jobs are displayed on DLI's SparkUI.
- A maximum of 1,000 job results can be displayed on the console. To view more or all jobs, export the job data to OBS.
- To export job run logs, you must have the permission to access OBS buckets. You need to configure a DLI job bucket on the **Global Configuration** > **Project** page in advance.
- **View Log** and **Export Log** buttons are not available for synchronization jobs and jobs running on the default queue.
- Only Spark jobs support custom images.
- An elastic resource pool supports a maximum of 32,000 CUs.

5.2 SQL Job Management

SQL jobs allow you to execute SQL statements entered in the **SQL job editing window**, import data, and export data.

SQL job management provides the following functions:

- **Searching for Jobs**: Search for jobs that meet the search criteria.
- **Viewing Job Details**: Display job details.
- **Terminating a Job**: Stop a job in the **Submitting** or **Running** status.
- **Exporting Query Results**: A maximum of 1000 records can be displayed in the query result on the console. To view more or all data, you can export the data to OBS.

SQL Jobs Page

On the **Overview** page of the DLI console, click **SQL Jobs** to go to the SQL job management page. Alternatively, you can click **Job Management** > **SQL Jobs**. The job list displays all SQL jobs. If there are a large number of jobs, they will be displayed on multiple pages. You can switch to the specified page as needed. DLI allows you to view jobs in all statuses. By default, jobs in the job list are displayed in descending order of the job creation time.

Table 5-1 SQL Job management parameters

Parameter	Description
Queues	Name of the queue to which a job belongs
Username	Name of the user who executed the job.
Type	Job type. The following types are supported: • **IMPORT**: A job that imports data to DLI • **EXPORT**: A job that exports data from DLI • **DCL**: Conventional DCLs and operations related to queue permissions • **DDL**:Conventional DDLs, including creating and deleting databases and tables • **QUERY**: A job that querys data by running SQL statements • **INSERT**: A job that inserts data by running SQL statements • **UPDATE**: A job that updates data. • **DELETE**: A job that deletes a SQL job. • **DATA_MIGRATION**: A job that migrates data. • **RESTART_QUEUE**: A job that restarts a queue. • **SCALE_QUEUE**: A job that changes queue specifications, including sale-out and scale-in.

Parameter	Description
Status	Job status. Possible values are as follows: • **Submitting** • **Running** • **Finished** • **Canceled** • **Failed** • **Scaling**
Query	SQL statements for operations such as exporting and creating tables You can click  to copy the query statement.
Duration	Running duration of a job
Created	Time when a job is created. Jobs can be displayed in ascending or descending order of the job creation time.

Parameter	Description
Operation	• **Edit**: Edit the job. • **Cancel** – You can terminate a job only when the job is in **Submitting** or **Running** status. – A job whose status is **Finished**, **Failed**, or **Canceled** cannot be terminated. – If the **Cancel** button is gray, you are not allowed to perform this operation. • **Re-execute**: Execute the job again. • **SparkUI**: Display the Spark job execution page. **NOTE** • When you execute a job on a created queue, the cluster is restarted. It takes about 10 minutes. If you click **SparkUI** before the cluster is created, an empty **projectID** will be cached. The SparkUI page cannot be displayed. You are advised to use a dedicated queue so that the cluster will not be released. Alternatively, wait for a while after the job is submitted (the cluster is created), and then check **SparkUI**. • Currently, only the latest 100 job information records are displayed on the SparkUI of DLI. • In addition to the preceding operations, the following operations are available for QUERY jobs and asynchronous DDL jobs. – **View Result**: View the job running result. – **Export Result**: Export the job running result to the created OBS bucket. For details, see **Exporting Query Results**. • In addition to the preceding operations, the EXPORT job also includes the following operations: – **Download** • **View Log**: Save job logs to the temporary OBS bucket created by DLI. • **Export Log**: Export logs to the created OBS bucket. If the job is in the **Running** state, logs cannot be exported. **NOTE** To export the logs, you need to obtain the permission to create an OBS bucket. Log archiving and export are not available for synchronization jobs and jobs running on the default queue.

Searching for a Job

On the **SQL Jobs** page, you can search jobs with any of the following operations.

- Select a queue name.

- Set the date range.
- Enter a username, statement, or job ID.
- Select the creation time in ascending or descending order.
- Select a job type.
- Select a job status.
- Select the job execution duration in ascending or descending order.

Viewing Job Details

On the **SQL Jobs** page, you can click ˅ in front of a job record to view details about the job.

Job details vary with job types. The job details vary depending on the job types, status, and configuration options. The following describes how to load data, create a table, and select a job. For details about other job types, see the information on the management console.

- **Load data** (job type: IMPORT) include the following information: queue, job ID, username, type, status, execution statement, running duration, creation time, end time, parameter settings, label, number of results, scanned data, number of scanned data, number of error records, storage path, data format, database, table, table header, separator, reference character, escape character, date format, timestamp format, total CPU used, and output bytes.
- **Create table** (job type: DDL) include the following information: queue, job ID, username, type, status, execution statement, running duration, creation time, end time, parameter settings, tags, number of results, scanned data, and database.
- **Select** (job type: QUERY) include the following information: queue, job ID, username, type, status, execution statement, running duration, creation time, end time, parameter setting, label, number of results (results of successful executions can be exported), and scanned data, username, result status (results of successful tasks can be viewed. Failure causes of failed tasks are displayed), database, total CPU used, and output bytes.

  NOTE

  - **Total CPU Used (Core x ms)**: total CPU used during job execution.
  - **Output Bytes**: number of output bytes after the job is executed.

Terminating a Job

On the **SQL Jobs** page, you can click **Terminate** in the **Operation** column to stop a submitting or running job.

Exporting Query Results

A maximum of 1000 records can be displayed in the query result on the console. To view more or all data, you can export the data to OBS. The procedure is as follows:

You can export results on the **SQL Jobs** page or the **SQL Editor** page.

- On the **Job Management** > **SQL Jobs** page, you can click **More** > **Export Result** in the **Operation** column to export the query result.
- After the query statements are successfully executed on the **SQL Editor** page, click  on the right of the **View Result** tab page to export the query result.

NOTE

If no column of the numeric type is displayed in the query result, the result cannot be exported.

Table 5-2 Exporting parameters

Parameter	Description
Data Format	Format of the exported query result file. This parameter can be set to **json** or **csv**.
Queues	The queue where the jobs are executed. SQL jobs can be executed only in SQL queues. For details about how to create a queue, see **Creating a Queue**.
Compression Format	Compression format of the data to be exported. The following options are supported: • none • bzip2 • deflate • gzip
Storage Path	OBS path to store the result. **NOTE** • After selecting an OBS bucket, enter a name for the folder. If the folder does not exist, it will be created in OBS. • The folder name cannot contain the special characters of \ / : * ? "< > |, and cannot start or end with a dot (.).
Export Mode	Mode for saving the exported query result data. • **New OBS directory**: If the specified export directory exists, an error is reported and the export operation cannot be performed. • **Existing OBS directory (Overwritten)**: If you create a file in the specified directory, the existing file will be overwritten.
Number of Results	Number of exported query results. If no value is entered or the value is **0**, all results are exported.
Table Header	Whether the data to be exported contains table headers.

5.3 Flink Job Management

5.3.1 Overview

On the Job Management page of Flink jobs, you can submit a Flink job. Currently, the following job types are supported:

- **Flink SQL** uses SQL statements to define jobs and can be submitted to any general purpose queue.
- **Flink Jar** customizes a JAR package job based on Flink APIs. It runs on dedicated queues.

Flink job management provides the following functions:

- **Managing Flink Job Permissions**
- **Creating a Flink SQL Job**
- **Creating a Flink Jar Job**
- **Debugging a Job**
- **Editing a job**
- **Starting a Job**
- **Stopping a Job**
- **Deleting a Job**
- **Exporting a Job**
- **Importing a Job**
- **Modifying Name and Description**
- **Importing to a Savepoint**
- **Triggering a Savepoint**
- **Runtime Configuration**
- **Job Details**
- **Tag Management**

Assigning Agency Permissions

Agencies are required for DLI to execute Flink jobs. You can set the agency when logging in to the management console for the first time or go to **Global Configurations** > **Service Authorization** to modify the agencies.

The permissions are as follows:

- **Tenant Administrator (global)** permissions are required to access data from OBS to execute Flink jobs on DLI, for example, obtaining OBS/GaussDB(DWS) data sources, log dump (including bucket authorization), checkpointing enabling, and job import and export.

  NOTE

  Due to cloud service cache differences, permission setting operations require about 60 minutes to take effect.

- **DIS Administrator** permissions are required to use DIS data as the data source of DLI Flink jobs.

  NOTE

  Due to cloud service cache differences, permission setting operations require about 30 minutes to take effect.

- To use CloudTable data as the data source of DLI Flink jobs, **CloudTable Administrator** permissions are required.

  NOTE

  Due to cloud service cache differences, permission setting operations require about 3 minutes to take effect.

Flink Jobs Page

On the **Overview** page, click **Flink Jobs** to go to the Flink job management page. Alternatively, you can choose **Job Management** > **Flink Jobs** from the navigation pane on the left. The page displays all Flink jobs. If there are a large number of jobs, they will be displayed on multiple pages. DLI allows you to view jobs in all statuses.

Table 5-3 Job management parameters

Parameter	Description
ID	ID of a submitted Flink job, which is generated by the system by default.
Name	Name of the submitted Flink job.
Type	Type of the submitted Flink job. Including: • Flink SQL: Flink SQL jobs • Flink Jar: Flink Jar jobs
Status	Job statuses, including: • **Draft** • **Submitting** • **Submission failed** • **Running**: After the job is submitted, a normal result is returned. • **Running exception**: The job stops running due to an exception. • **Downloading** • **Idle** • **Stopping** • **Stopped** • **Stopping failed** • **Creating the savepoint** • **Completed**

Parameter	Description
Description	Description of the submitted Flink job.
Username	Name of the user who submits a job.
Created	Time when a job is created.
Started	Time when a Flink job starts to run.
Duration	Time consumed by job running.
Operation	• **Edit**: Edit a created job. For details, see **Editing a Job**. • **Start**: Start and run a job. For details, see **Starting a Job**. • **More** – **FlinkUI**: After you click this button, the Flink job execution page is displayed. **NOTE** When you execute a job on a created queue, the cluster is restarted. It takes about 10 minutes. If you click **FlinkUI** before the cluster is created, an empty **projectID** will be cached. The **FlinkUI** page cannot be displayed. You are advised to use a dedicated queue so that the cluster will not be released. Alternatively, wait for a while after the job is submitted (the cluster is created), and then check **FlinkUI**. – **Stop**: Stop a Flink job. If this function is unavailable, jobs in the current status cannot be stopped. – **Delete**: Delete a job. **NOTE** A deleted job cannot be restored. – **Modify Name and Description**: You can modify the name and description of a job. For details, see **Modifying Name and Description**. – **Import Savepoint**: Import the data exported from the original CS job. For details, see **Importing to a Savepoint**. – **Trigger Savepoint**: You can click this button for jobs in the **Running** status to save the job status. For details, see **Triggering a Savepoint**. – **Permissions**: You can view the user permissions corresponding to the job and grant permissions to other users. For details, see **Managing Flink Job Permissions**. – **Runtime Configuration**: You can enable **Alarm Generation upon Job Exception** and **Auto Restart upon Exception**. For details, see **Runtime Configuration**.

5.3.2 Managing Flink Job Permissions

Scenario

- You can isolate Flink jobs allocated to different users by setting permissions to ensure data query performance.
- The administrator and job creator have all permissions, which cannot be set or modified by other users.

Flink Job Permission Operations

1. On the left of the DLI management console, choose **Job Management** > **Flink Jobs**.
2. Select the job to be configured and choose **More** > **Permissions** in the **Operation** column. The **User Permissions** area displays the list of users who have permissions on the job.

   You can assign queue permissions to new users, modify permissions for users who have some permissions of a queue, and revoke all permissions of a user on a queue.

   – Assign permissions to a new user.

     A new user does not have permissions on the job.

     i. Click **Grant Permission** on the right of **User Permissions** page. The **Grant Permission** dialog box is displayed.

     ii. Specify **Username** and select corresponding permissions.

     iii. Click **OK**.

     Table 5-4 describes the related parameters.

     Table 5-4 Permission parameters

Parameter	Description
Username	Name of the user you want to grant permissions to. **NOTE** The username is the name of an existing IAM user. In addition, the user can perform authorization operations only after logging in to the platform.

Parameter	Description
Permissions to be granted to the user	• **Select all**: All permissions are selected. • **View Job Details**: This permission allows you to view the job details. • **Modify Job**: This permission allows you to modify the job. • **Delete Job**: This permission allows you to delete the job. • **Start Job**: This permission allows you to start the job. • **Stop Job**: This permission allows you to stop the job. • **Export Job**: This permission allows you to export the job. • **Grant Permission**: This permission allows you to grant job permissions to other users. • **Revoke Permission**: This permission allows you to revoke the job permissions that other users have but cannot revoke the job creator's permissions. • **View Other User's Permissions**: This permission allows you to view the job permissions of other users.

  - To assign or revoke permissions of a user who has some permissions on the job, perform the following steps:
    i. In the list under **User Permissions** for a job, select the user whose permissions need to be modified and click **Set Permission** in the **Operation** column.
    ii. In the displayed **Set Permission** dialog box, modify the permissions of the current user. **Table 5-4** lists the detailed permission descriptions.

    If all options under **Set Permission** are gray, you are not allowed to change permissions on this job. You can apply to the administrator, job creator, or other authorized users for job permission granting and revoking.
    iii. Click **OK**.
  - To revoke all permissions of a user on a job, perform the following steps:

    In the list under **User Permissions** for a job, locate the user whose permissions need to be revoked, click **Revoke Permission** in the **Operation** column, and click **Yes**. After this operation, the user does not have any permission on the job.

Flink Job Permissions

- **View Job Details**
  - Tenants and the admin user can view and operate all jobs.

  - Subusers and users with the read-only permission can only view their own jobs.

    NOTE

    If another user grants any permission other than the job viewing permission to a subuser, the job is displayed in the job list, but the details cannot be viewed by the subuser.

- **Start Job**
  - To use a dedicated queue, you must have the permission to submit and start jobs.
  - To use a shared queue, you only need to have the permission to start jobs.
- **Stop Job**
  - To use a dedicated queue, you must have the permission to stop jobs and queues.
  - To use a shared queue, you only need to have the permission to stop jobs.
- **Delete Job**
  - If a job can be deleted, you can delete the job if you were granted this permission.
  - If a job cannot be deleted, the system stops the job before you delete it. For details about how to stop a job, see **Stop Job**. In addition, you must have the permission to delete the job.
- **Create Job**
  - By default, sub-users cannot create jobs.
  - To create a job, you must have this permission. Currently, only the admin user has the permission to create jobs. In addition, the user must have the permission of the related package group or package used by the job.
- **Modify Job**

  When modifying a job, you need to have the permission to update the job and the permission to the package group or package used by the job belongs.

5.3.3 Preparing Flink Job Data

To create a Flink job, you need to enter the data source and data output channel, that is, source and sink. To use another service as the source or sink stream, you need to apply for the service first.

Flink jobs support the following data sources and output channels:

- DIS as the data input and output channel

  To use DIS as the data source and output channel, you need to enable DIS first.

  For details about how to create a DIS stream, see **Creating a DIS Stream** in the *Data Ingestion Service User Guide*.

  After applying for a DIS stream, you can upload local data to DIS to provide data sources for Flink jobs in real time. For details, see **Sending Data to DIS** in the *Data Ingestion Service User Guide*.

  An example is provided as follows:

```
1,lilei,bmw320i,28
2,hanmeimei,audia4,27
```

- OBS as the data source

  To use OBS as the data source, enable OBS first. For details about how to enable OBS, see **Enabling OBS** in the *Object Storage Service Console Operation Guide*.

  After you enable OBS, upload local files to OBS using the Internet. For detailed operations, see **Uploading a File** in the *Object Storage Service Console Operation Guide*.

- RDS as the output channel

  To use RDS as the output channel, create an RDS instance. For details, see **Creating a DB Instance** in the *Relational Database Service User Guide*.

- SMN as the output channel

  To use SMN as the output channel, create an SMN topic to obtain the URN resource ID and then add topic subscription. For detailed operations, see **Getting Started** in the *Simple Message Notification User Guide*.

- Kafka as the data input and output channel

  If Kafka serves as both the source and sink streams, create an enhanced datasource connection between Flink jobs and Kafka. For details, see **Enhanced Datasource Connections**.

  If the port of the Kafka server is listened on by the host name, you need to add the mapping between the host name and IP address of the Kafka Broker node to the datasource connection.

- CloudTable as the data input and output channel

  To use CloudTable as the data input and output channel, create a cluster in CloudTable and obtain the cluster ID.

- CSS as the output channel

  To use CSS as the data output channel, create a cluster in CSS and obtain the cluster's private network address. For details, see **Getting Started** in the *Cloud Search Service User Guide*.

- DCS as the output channel

  To use DCS as the output channel, create a Redis cache instance in DCS and obtain the address used for Flink jobs to connect to the Redis instance. For detailed operations, see **Getting Started** in the *Distributed Cache Service User Guide*.

5.3.4 Creating a Flink SQL Job

This section describes how to create a Flink SQL job. You can use Flink SQLs to develop jobs to meet your service requirements. Using SQL statements simplifies logic implementation. You can edit Flink SQL statements for your job in the DLI SQL editor. This section describes how to use the SQL editor to write Flink SQL statements.

Prerequisites

- You have prepared the data input and data output channels. For details, see **Preparing Flink Job Data**.

- When you use a Flink SQL job to access other external data sources, such as OpenTSDB, HBase, Kafka, DWS, RDS, CSS, CloudTable, DCS Redis, and DDS MongoDB, you need to create a cross-source connection to connect the job running queue to the external data source.
  - For details about the external data sources that can be accessed by Flink jobs, see **Cross-Source Analysis Development Methods**.
  - For details about how to create a datasource connection, see **Enhanced Datasource Connections**. After a datasource connection is created, you can choose **More** > **Test Address Connectivity** in the **Operation** column on the **Queue Management** page to check whether the network connection between the queue and the external data source is normal. For details, see **Testing Address Connectivity**.

Creating a Flink SQL Job

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 In the upper right corner of the **Flink Jobs** page, click **Create Job**.

Step 3 Specify job parameters.

Table 5-5 Job configuration information

Parameter	Description
Type	Set **Type** to **Flink SQL**. You will need to rewrite SQL statements to start the job.
Name	Name of a job. Enter 1 to 57 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed. **NOTE** The job name must be globally unique.
Description	Description of a job. It can contain a maximum of 512 characters.
Template Name	You can select a sample template or a custom job template. For details about templates, see **Managing Flink Templates**.

Parameter	Description
Tag	Tags used to identify cloud resources. A tag includes the tag key and tag value. If you want to use the same tag to identify multiple cloud resources, that is, to select the same tag from the drop-down list box for all services, you are advised to create predefined tags on the Tag Management Service (TMS). **NOTE** • A maximum of 20 tags can be added. • Only one tag value can be added to a tag key. • The key name in each resource must be unique. • Tag key: Enter a tag key name in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with **_sys_**. • Tag value: Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 225 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

Step 4 Click **OK** to enter the editing page.

Step 5 Edit a Flink SQL job.

Enter details SQL statements in the statement editing area. For details about SQL syntax, see the Data Lake Insight SQL Syntax Reference.

Step 6 Click **Check Semantics**.

- You can **Debug** or **Start** a job only after the semantic verification is successful.
- If verification is successful, the message "The SQL semantic verification is complete. No error." will be displayed.
- If verification fails, a red "X" mark will be displayed in front of each SQL statement that produced an error. You can move the cursor to the "X" mark to view error details and change the SQL statement as prompted.

Step 7 Set job running parameters.

Table 5-6 Running parameters

Parameter	Description
Queue	A shared queue is selected by default. You can select a custom queue as needed. **NOTE** • During job creation, a sub-user can only select a queue that has been allocated to the user. • If the remaining capacity of the selected queue cannot meet the job requirements, the system automatically scales up the capacity. When a queue is idle, the system automatically scales in the queue.
UDF Jar	If you selected custom queues, you need to configure this parameter. You can customize a UDF Jar file. Before you select a JAR file, upload the corresponding JAR package to the OBS bucket and choose **Data Management > Package Management** to create a package. For details, see **Creating a Package**. In SQL, you can call a user-defined function that is inserted into a JAR file.
CUs	Sum of the number of compute units and job manager CUs of DLI. One CU equals 1 vCPU and 4 GB. The configured number of CUs is the number of CUs required for job running and cannot exceed the number of CUs in the bound queue.
Job Manager CUs	Number of CUs of the management unit.
Parallelism	Number of Flink SQL jobs that run at the same time Properly increasing the number of parallel threads improves the overall computing capability of the job. However, the switchover overhead caused by the increase of threads must be considered. **NOTE** • This value cannot be greater than four times the compute units (number of CUs minus the number of job manager CUs). • The priority of the number of parallel tasks on this page is lower than that set in the code.
Task Manager Configuration	Whether to set Task Manager resource parameters If this option is selected, you need to set the following parameters: • **CU(s) per TM**: Number of resources occupied by each Task Manager. • **Slot(s) per TM**: Number of slots contained in each Task Manager.

Parameter	Description
OBS Bucket	OBS bucket to store job logs and checkpoint information. If the selected OBS bucket is not authorized, click **Authorize**. **NOTE** If **Enable Checkpointing** and **Save Job Log** are both selected, you only need to authorize OBS once.
Save Job Log	Whether to save the job running logs to OBS The logs are saved in the following path: *Bucket name***/jobs/logs/***Directory starting with the job ID*. To go to this path, go to the job list and click the job name. On the **Run Log** tab page, click the provided OBS link. **CAUTION** You are advised to select this parameter. Otherwise, no run log is generated after the job is executed. If the job is abnormal, the run log cannot be obtained for fault locating. If this option is selected, you need to set the following parameters: **OBS Bucket**: Select an OBS bucket to store user job logs. If the selected OBS bucket is not authorized, click **Authorize**. **NOTE** If **Enable Checkpointing** and **Save Job Log** are both selected, you only need to authorize OBS once.
Alarm Generation upon Job Exception	Whether to report job exceptions, for example, abnormal job running or exceptions due to an insufficient balance, to users via SMS or email If this option is selected, you need to set the following parameters: **SMN Topic** Select a user-defined SMN topic. For details about how to create a custom SMN topic, see "Creating a Topic" in *Simple Message Notification User Guide*.

Parameter	Description
Enable Checkpointing	Whether to enable job snapshots. If this function is enabled, jobs can be restored based on the checkpoints. If this option is selected, you need to set the following parameters: • **Checkpoint Interval** indicates the interval for creating checkpoints. The value ranges from 1 to 999999, and the default value is **30**. • **Checkpoint Mode** can be set to either of the following values: – **At least once**: Events are processed at least once. – **Exactly once**: Events are processed only once. • **OBS Bucket**: Select an OBS bucket to store your checkpoints. If the selected OBS bucket is not authorized, click **Authorize**. The checkpoint path is *Bucket name***/jobs/checkpoint/***Directory starting with the job ID*. **NOTE** If **Enable Checkpointing** and **Save Job Log** are both selected, you only need to authorize OBS once.
Auto Restart upon Exception	Whether to enable automatic restart. If this function is enabled, any job that has become abnormal will be automatically restarted. If this option is selected, you need to set the following parameters: • **Max. Retry Attempts**: maximum number of retry times upon an exception. The unit is times/hour. – **Unlimited**: The number of retries is unlimited. – **Limited**: The number of retries is user-defined. • **Restore Job from Checkpoint**: This parameter is available only when **Enable Checkpointing** is selected.
Idle State Retention Time	Defines for how long the state of a key is retained without being updated before it is removed in **GroupBy** or **Window**. The default value is 1 hour.
Dirty Data Policy	Select a policy for processing dirty data. The following policies are supported: **Ignore**, **Trigger a job exception**, and **Save**. **NOTE** **Save** indicates that the dirty data is stored to the OBS bucket selected above.
Dirty Data Dump Address	Set this parameter when **Dirty Data Policy** is set to **Save**. Click the address box to select the OBS path for storing dirty data.

Step 8 (Optional) Debug parameters as required. The job debugging function is used only to verify the SQL logic and does not involve data write operations. For details, see **Debugging a Flink Job**.

Step 9 (Optional) Set the runtime configuration as required.

Step 10 Click **Save**.

Step 11 Click **Start**. On the displayed **Start Flink Jobs** page, confirm the job specifications, and click **Start Now** to start the job.

After the job is started, the system automatically switches to the **Flink Jobs** page, and the created job is displayed in the job list. You can view the job status in the **Status** column. After a job is successfully submitted, the job status will change from **Submitting** to **Running**. After the execution is complete, the message **Completed** is displayed.

If the job status is **Submission failed** or **Running exception**, the job submission failed or the job did not execute successfully. In this case, you can move the cursor over the status icon in the **Status** column of the job list to view the error details. You can click to copy error information. After handling the fault based on the provided information, resubmit the job.

NOTE

Other available buttons are as follows:

- **Save As**: Save the created job as a new job.
- **Debug**: Perform job debugging. For details, see **Debugging a Flink Job**.
- **Format**: Format the SQL statements in the editing box.
- **Set as Template**: Set the created SQL statements as a job template.
- **Theme Settings**: Set the theme related parameters, including **Font Size**, **Wrap**, and **Page Style**.

----End

5.3.5 Creating a Flink Jar Job

This section describes how to create a Flink Jar job. You can perform secondary development based on Flink APIs, build your own JAR file, and submit the JAR file to DLI queues. DLI is fully compatible with open-source community APIs. To create a custom Flink job, you need to compile and build application JAR files. You must have a certain understanding of Flink secondary development and have high requirements related to stream computing complexity.

Prerequisites

- Ensure that a dedicated queue has been created. To create a dedicated queue, select **Dedicated Resource Mode** when you choose the type of a queue during purchase.
- When creating a Flink Jar job to access other external data sources, such as OpenTSDB, HBase, Kafka, GaussDB(DWS), RDS, CSS, CloudTable, DCS Redis, and DDS MongoDB, you need to create a cross-source connection to connect the job running queue to the external data source.
  - For details about the external data sources that can be accessed by Flink jobs, see **Cross-Source Analysis Development Methods**.

  - For details about how to create a datasource connection, see **Enhanced Datasource Connections**.

    On the **Resources** > **Queue Management** page, locate the queue you have created, and choose **More** > **Test Address Connectivity** in the **Operation** column to check whether the network connection between the queue and the data source is normal. For details, see **Testing Address Connectivity**.

- When running a Flink Jar job, you need to build the secondary development application code into a Jar package and upload the JAR package to the created OBS bucket. Choose **Data Management** > **Package Management** to create a package. For details, see **Creating a Package**.

  NOTE

  DLI does not support the download function. If you need to modify the uploaded data file, please edit the local file and upload it again.

- Flink dependencies have been built in the DLI server and security hardening has been performed based on the open-source community version. To avoid dependency package compatibility issues or log output and dump issues, be careful to exclude the following files when packaging:
  - Built-in dependencies (or set the package dependency scope to **provided** in Maven or SBT)
  - Log configuration files (example, **log4j.properties**/**logback.xml**)
  - JAR package for log output implementation (example, **log4j**).

Creating a Flink Jar Job

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 In the upper right corner of the **Flink Jobs** page, click **Create Job**.

Step 3 Specify job parameters.

Table 5-7 Job configuration information

Parameter	Description
Type	Select **Flink Jar**.
Name	Name of a job. Enter 1 to 57 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed. **NOTE** The job name must be globally unique.
Description	Description of a job. It can be up to 512 characters long.

Parameter	Description
Tag	Tags used to identify cloud resources. A tag includes the tag key and tag value. If you want to use the same tag to identify multiple cloud resources, that is, to select the same tag from the drop-down list box for all services, you are advised to create predefined tags on the Tag Management Service (TMS). For details, see . **NOTE** • A maximum of 20 tags can be added. • Only one tag value can be added to a tag key. • The key name in each resource must be unique. • Tag key: Enter a tag key name in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with **_sys_**. • Tag value: Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 225 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

Step 4 Click **OK** to enter the editing page.

Step 5 Select a queue. Flink Jar jobs can run only on dedicated queues.

NOTE

- A Flink Jar job can run only on a pre-created dedicated queue.
- If no dedicated queue is available in the **Queue** drop-down list, create a dedicated queue and bind it to the current user.

Step 6 Configuring Flink Jar Job parameters

Table 5-8 Parameter description

Name	Description
Queue	A shared queue is selected by default. You can select a custom queue as needed.
Application	User-defined package. Before selecting a JAR file to be inserted, upload the corresponding JAR file to the OBS bucket and choose **Data Management** > **Package Management** to create a package. For details, see **Creating a Package**. For details about built-in dependency packages, see **Built-in Dependencies**.

Name	Description
Main Class	The name of the JAR package to be loaded, for example, **KafkaMessageStreaming**. • **Default**: Specified based on the **Manifest** file in the JAR package. • **Manually assign**: You must enter the class name and confirm the class arguments (separate arguments with spaces). **NOTE** When a class belongs to a package, the main class path must contain the complete package path, for example, **packagePath.KafkaMessageStreaming**.
Class Arguments	List of arguments of a specified class. The arguments are separated by spaces. Flink parameters support replacement of non-sensitive global variables. For example, if you add the global variable **windowsize** in **Global Configuration** > **Global Variables**, you can add the **-windowsSize {{windowsize}}** parameter for the Flink Jar job.
JAR Package Dependencies	Select a user-defined package dependency. The dependent program packages are stored in the classpath directory of the cluster. Before selecting a JAR file to be inserted, upload the corresponding JAR file to the OBS bucket and choose **Data Management > Package Management** to create a package. Select **JAR** as the package type. For details, see **Creating a Package**. For details about built-in dependency packages, see **Built-in Dependencies**.
Other Dependencies	User-defined dependency files. Other dependency files need to be referenced in the code. Before selecting a dependency file, upload the file to the OBS bucket and choose **Data Management > Package Management** to create a package. The package type is not limited. For details, see **Creating a Package**. You can add the following command to the application to access the corresponding dependency file. In the command, **fileName** indicates the name of the file to be accessed, and **ClassName** indicates the name of the class that needs to access the file. `ClassName.class.getClassLoader().getResource("userData/fileName")`
Flink Version	Before selecting a Flink version, you need to select the queue to which the Flink version belongs.

Name	Description
Runtime Configuration	User-defined optimization parameters. The parameter format is **key=value**. Flink optimization parameters support replacement non-sensitive global variable. For example, if you create global variable **phase** in **Global Configuration** > **Global Variables**, optimization parameter **table.optimizer.agg-phase.strategy={{phase}}** can be added to the Flink Jar job.

Step 7 Configure job parameters.

Table 5-9 Parameter description

Name	Description
CUs	One CU has one vCPU and 4-GB memory. The number of CUs ranges from 2 to 10,000.
Job Manager CUs	Set the number of CUs on a management unit. The value ranges from 1 to 4. The default value is 1.
Parallelism	Maximum number of parallel operators in a job. The value ranges from 1 to 10,000. **NOTE** • The value must be less than or equal to four times the number of compute units (CUs minus the number of job manager CUs). • You are advised to set this parameter to a value greater than that configured in the code. Otherwise, job submission may fail.
Task Manager Configuration	Whether to set Task Manager resource parameters If this option is selected, you need to set the following parameters: • **CU(s) per TM**: Number of resources occupied by each Task Manager. • **Slot(s) per TM**: Number of slots contained in each Task Manager.
Save Job Log	Whether to save the job running logs to the OBS bucket. **CAUTION** You are advised to select this parameter. Otherwise, no run log is generated after the job is executed. If the job is abnormal, the run log cannot be obtained for fault locating. If this option is selected, you need to set the following parameters: **OBS Bucket**: Select an OBS bucket to store job logs. If the selected OBS bucket is not authorized, click **Authorize**.

Name	Description
Alarm Generation upon Job Exception	Whether to report job exceptions, for example, abnormal job running or exceptions due to an insufficient balance, to users via SMS or email If this option is selected, you need to set the following parameters: **SMN Topic** Select a user-defined SMN topic. For details about how to customize SMN topics, see **Creating a Topic** in the *Simple Message Notification User Guide*.
Auto Restart upon Exception	Whether to enable automatic restart. If this function is enabled, any job that has become abnormal will be automatically restarted. If this option is selected, you need to set the following parameters: • **Max. Retry Attempts**: maximum number of retry times upon an exception. The unit is **Times/hour**. – **Unlimited**: The number of retries is unlimited. – **Limited**: The number of retries is user-defined. • **Restore Job from Checkpoint**: Restore the job from the saved checkpoint. If you select this parameter, you also need to set **Checkpoint Path**. **Checkpoint Path**: Select the checkpoint saving path. The checkpoint path must be the same as that you set in the application package. Note that the checkpoint path for each job must be unique. Otherwise, the checkpoint cannot be obtained.

Step 8 Click **Save** on the upper right of the page.

Step 9 Click **Start** on the upper right side of the page. On the displayed **Start Flink Job** page, confirm the job specifications, and click **Start Now** to start the job.

After the job is started, the system automatically switches to the **Flink Jobs** page, and the created job is displayed in the job list. You can view the job status in the **Status** column. After a job is successfully submitted, the job status will change from **Submitting** to **Running**. After the execution is complete, the message **Completed** is displayed.

If the job status is **Submission failed** or **Running exception**, the job submission failed or the job did not execute successfully. In this case, you can move the cursor over the status icon in the **Status** column of the job list to view the error details. You can click to copy these details. After handling the fault based on the provided information, resubmit the job.

NOTE

Other available buttons are as follows:

Save As: Save the created job as a new job.

----End

5.3.6 Debugging a Flink Job

The job debugging function helps you check the logic correctness of your compiled SQL statements before running a job.

NOTE

- Currently, only Flink SQL jobs support this function.
- The job debugging function is used only to verify the SQL logic and does not involve data write operations.

Procedures

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 In the **Operation** column of the created Flink SQL job, click **Edit**. The page for editing the Flink SQL job is displayed.

For a job that is being created, you can debug the job on the editing page.

Step 3 Click **Debug** above the SQL editing box to parse the edited SQL statements. The **Debugging Parameters** tab is displayed on the right of the page.

- **Dump Bucket**: Select an OBS bucket to save debugging logs. If you select an unauthorized OBS bucket, click **Authorize**.
- **Data Input Mode**: You can select CSV data stored in the OBS bucket or manually enter the data.
  - **OBS (CSV)**

    If you select this value, prepare OBS data first before using DLI. For details, see **Preparing Flink Job Data**. OBS data is stored in CSV format, where multiple records are separated by line breaks and different fields in a single record are separated by commas (,). In addition, you need to select a specific object in OBS as the input source data.

  - **Manual typing**

    If you select this value, compile SQL statements as data sources. In this mode, you need to enter the value of each field in a single record.

Step 4 Click **Start Debugging**. Once debugging is complete, the **Debugging Result** page appears.

- If the debugging result meets the expectation, the job is running properly.
- If the debugging result does not meet the expectation, business logic errors may have occurred. In this case, modify SQL statements and conduct debugging again.

----End

5.3.7 Performing Operations on a Flink Job

After a job is created, you can perform operations on the job as required.

- Editing a Job
- Starting a Job
- Stopping a Job
- Deleting a Job
- Exporting a Job
- Importing a Job
- Modifying Name and Description
- Importing to a Savepoint
- Triggering a Savepoint
- Runtime Configuration

Editing a Job

You can edit a created job, for example, by modifying the SQL statement, job name, job description, or job configurations.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 In the row where the job you want to edit locates, click **Edit** in the **Operation** column to switch to the editing page.

Step 3 Edit the job as required.

For details about how to edit a Flink SQL job, see Step 5 to Step 7 in **Creating a Flink SQL Job**.

For details about how to edit a user-defined Flink job, see Step 5 to Step 7 in **Creating a Flink Jar Job**.

----End

Starting a Job

You can start a saved or stopped job.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Use either of the following methods to start jobs:

- Starting a single job

  Select a job and click **Start** in the **Operation** column.

  Alternatively, you can select the row where the job you want to start locates and click **Start** in the upper left of the job list.

- Batch starting jobs

  Select the rows where the jobs you want to start locate and click **Start** in the upper left of the job list.

After you click **Start**, the **Start Flink Jobs** page is displayed.

Step 3 On the **Start Flink Jobs** page, confirm the job information. If they are correct, click **Start Now**.

After a job is started, you can view the job execution result in the **Status** column.

----End

Stopping a Job

You can stop a job in the **Running** or **Submitting** state.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Stop a job using either of the following methods:

- Stopping a job

  Locate the row that contains the job to be stopped, click **More** in the **Operation** column, and select **Stop**.

  Alternatively, you can select the row where the job you want to stop locates and click **Stop** in the upper left of the job list.

- Batch stopping jobs

  Locate the rows containing the jobs you want to stop and click **Stop** in the upper left of the job list.

Step 3 In the displayed **Stop Job** dialog box, click **OK** to stop the job.

NOTE

- Before stopping a job, you can trigger a savepoint to save the job status information. When you start the job again, you can choose whether to restore the job from the savepoint.
- If you select **Trigger savepoint**, a savepoint is created. If **Trigger savepoint** is not selected, no savepoint is created. By default, the savepoint function is disabled.
- The lifecycle of a savepoint starts when the savepoint is triggered and stops the job, and ends when the job is restarted. The savepoint is automatically deleted after the job is restarted.

When a job is being stopped, the job status is displayed in the **Status** column of the job list. The details are as follows:

- **Stopping**: indicates that the job is being stopped.
- **Stopped**: indicates that the job is stopped successfully.
- **Stop failed**: indicates that the job failed to be stopped.

----End

Deleting a Job

If you do not need to use a job, perform the following operations to delete it. A deleted job cannot be restored. Therefore, exercise caution when deleting a job.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Perform either of the following methods to delete jobs:

- Deleting a single job

  Locate the row containing the job you want to delete and click **More** > **Delete** in the **Operation** column.

Alternatively, you can select the row containing the job you want to delete and click **Delete** in the upper left of the job list.

- Deleting jobs in batches

  Select the rows containing the jobs you want to delete and click **Delete** in the upper left of the job list.

Step 3 Click **Yes**.

----End

Exporting a Job

You can export the created Flink jobs to an OBS bucket.

This mode is applicable to the scenario where a large number of jobs need to be created when you switch to another region, project, or user. In this case, you do not need to create a job. You only need to export the original job, log in to the system in a new region or project, or use a new user to import the job.

NOTE

When switching to another project or user, you need to grant permissions to the new project or user. For details, see **Managing Flink Job Permissions**.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click **Export Job** in the upper right corner. The **Export Job** dialog box is displayed.

Step 3 Select the OBS bucket where the job is stored. Click **Next**.

Step 4 Select job information you want to export.

By default, configurations of all jobs are exported. You can enable the **Custom Export** function to export configurations of the desired jobs.

Step 5 Click **Confirm** to export the job.

----End

Importing a Job

You can import the Flink job configuration file stored in the OBS bucket to the **Flink Jobs** page of DLI.

This mode is applicable to the scenario where a large number of jobs need to be created when you switch to another region, project, or user. In this case, you do not need to create a job. You only need to export the original job, log in to the system in a new region or project, or use a new user to import the job.

If you need to import a self-created job, you are advised to use the job creation function. For details, see **Creating a Flink SQL Job** and **Creating a Flink Jar Job**.

NOTE

- When switching to another project or user, you need to grant permissions to the new project or user. For details, see **Managing Flink Job Permissions**.
- Only jobs whose data format is the same as that of Flink jobs exported from DLI can be imported.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click **Import Job** in the upper right corner. The **Import Job** dialog box is displayed.

Step 3 Select the complete OBS path of the job configuration file to be imported. Click **Next**.

Step 4 Configure the same-name job policy and click next. Click **Next**.

- Select **Overwrite job of the same name**. If the name of the job to be imported already exists, the existing job configuration will be overwritten and the job status switches to **Draft**.
- If **Overwrite job of the same name** is not selected and the name of the job to be imported already exists, the job will not be imported.

Step 5 Ensure that **Config File** and **Overwrite Same-Name Job** are correctly configured. Click **Confirm** to import the job.

----End

Modifying Name and Description

You can change the job name and description as required.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 In the **Operation** column of the job whose name and description need to be modified, choose **More > Modify Name and Description**. The **Modify Name and Description** dialog box is displayed. Change the name or modify the description of a job.

Step 3 Click **OK**.

----End

Triggering a Savepoint

When you need to stop a job, you can create a savepoint to save the job status information. In this case, when you restart the job, you can choose to restore the job from the latest savepoint.

NOTE

- You can click **Trigger Savepoint** for jobs in the **Running** status to save the job status.
- The lifecycle of a savepoint starts when the savepoint is triggered and stops the job, and ends when the job is restarted. The savepoint is automatically deleted after the job is restarted.

Importing to a Savepoint

You can import a savepoint to restore the job status. For details about the savepoint, see **Checkpointing** at the official website of Flink.

You need to select the OBS path of the save point.

Runtime Configuration

You can select **Runtime Configuration** to configure job exception alarms and restart options.

NOTE

Flink SQL jobs and Flink Jar jobs are supported.

1. In the **Operation** column of the Flink job, choose **More > Runtime Configuration**.
2. In the **Runtime Configuration** dialog box, set the following parameters:

Table 5-10 Running parameters

Parameter	Description
Name	Job name.
Alarm Generation upon Job Exception	Whether to report job exceptions, for example, abnormal job running or exceptions due to an insufficient balance, to users via SMS or email. If this option is selected, you need to set the following parameters: **SMN Topic** Select a user-defined SMN topic. For details about how to customize SMN topics, see **Creating a Topic** in the *Simple Message Notification User Guide*.

Parameter	Description
Auto Restart upon Exception	Whether to enable automatic restart. If this function is enabled, any job that has become abnormal will be automatically restarted. If this option is selected, you need to set the following parameters: • **Max. Retry Attempts**: maximum number of retry times upon an exception. The unit is times/hour. – **Unlimited**: The number of retries is unlimited. – **Limited**: The number of retries is user-defined. • **Restore Job from Checkpoint**: Restore the job from the saved checkpoint. **NOTE** For Flink streaming SQL jobs, you need to select **Enable Checkpoint** on the job editing page before configuring this parameter. If this parameter is selected, you need to set **Checkpoint Path** for Flink Jar jobs. **Checkpoint Path**: Select the checkpoint saving path. The checkpoint path must be the same as that you set in the application package. Note that the checkpoint path for each job must be unique. Otherwise, the checkpoint cannot be obtained.

5.3.8 Flink Job Details

After creating a job, you can view the job details to learn about the following information:

- **Viewing Job Details**
- **Checking the Job Monitoring Information**
- **Viewing the Task List of a Job**
- **Viewing the Job Execution Plan**
- **Viewing Job Submission Logs**
- **Viewing Job Running Logs**

Viewing Job Details

This section describes how to view job details. After you create and save a job, you can click the job name to view job details, including SQL statements and parameter settings. For a Jar job, you can only view its parameter settings.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click the name of the job to be viewed. The **Job Detail** tab is displayed.

In the **Job Details** tab, you can view SQL statements, configured parameters.

The following uses a Flink SQL job as an example.

Table 5-11 Description

Parameter	Description
Type	Job type, for example, **Flink SQL**
Name	Flink job name
Description	Description of a Flink job
Status	Running status of a job
Running Mode	If your job runs on a shared queue, this parameter is **Shared**. If your job runs on a custom queue with dedicated resources, this parameter is **Exclusive**.
Flink Version	Version of Flink selected for the job.
Runtime Configuration	Displayed when a user-defined parameter is added to a job
CUs	Number of CUs configured for a job
Job Manager CUs	Number of job manager CUs configured for a job.
Parallelism	Number of jobs that can be concurrently executed by a Flink job
CU(s) per TM	Number of CUs occupied by each Task Manager configured for a job
Slot(s) per TM	Number of Task Manager slots configured for a job
OBS Bucket	OBS bucket name. After **Enable Checkpointing** and **Save Job Log** are enabled, checkpoints and job logs are saved in this bucket.
Save Job Log	Whether the job running logs are saved to OBS
Alarm Generation upon Job Exception	Whether job exceptions are reported
SMN Topic	Name of the SMN topic. This parameter is displayed when **Alarm Generation upon Job Exception** is enabled.
Auto Restart upon Exception	Whether automatic restart is enabled.
Max. Retry Attempts	Maximum number of retry times upon an exception. **Unlimited** means the number is not limited.
Savepoint	OBS path of the savepoint

Parameter	Description
Enable Checkpointing	Whether checkpointing is enabled
Checkpoint Interval	Interval between storing intermediate job running results to OBS. The unit is second.
Checkpoint Mode	Checkpoint mode. Available values are as follows: • **At least once**: Events are processed at least once. • **Exactly once**: Events are processed only once.
Idle State Retention Time	Defines for how long the state of a key is retained without being updated before it is removed in GroupBy or Window.
Dirty Data Policy	Policy for processing dirty data. The value is displayed only when there is a dirty data policy. Available values are as follows: **Ignore** **Trigger a job exception** **Save**
Dirty Data Dump Address	OBS path for storing dirty data when **Dirty Data Policy** is set to **Save**.
Created	Time when a job is created
Updated	Time when a job was last updated

----End

Checking the Job Monitoring Information

You can use Cloud Eye to view details about job data input and output.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click the name of the job you want. The job details are displayed.

Click **Job Monitoring** in the upper right corner of the page to switch to the Cloud Eye console.

The following table describes monitoring metrics related to Flink jobs.

Table 5-12 Monitoring metrics related to Flink jobs

Name	Description
Flink Job Data Read Rate	Displays the data input rate of a Flink job for monitoring and debugging. Unit: record/s.
Flink Job Data Write Rate	Displays the data output rate of a Flink job for monitoring and debugging. Unit: record/s.

Name	Description
Flink Job Total Data Read	Displays the total number of data inputs of a Flink job for monitoring and debugging. Unit: records
Flink Job Total Data Write	Displays the total number of output data records of a Flink job for monitoring and debugging. Unit: records
Flink Job Byte Read Rate	Displays the number of input bytes per second of a Flink job. Unit: byte/s
Flink Job Byte Write Rate	Displays the number of output bytes per second of a Flink job. Unit: byte/s
Flink Job Total Read Byte	Displays the total number of input bytes of a Flink job. Unit: byte
Flink Job Total Write Byte	Displays the total number of output bytes of a Flink job. Unit: byte
Flink Job CPU Usage	Displays the CPU usage of Flink jobs. Unit: %
Flink Job Memory Usage	Displays the memory usage of Flink jobs. Unit: %
Flink Job Max Operator Latency	Displays the maximum operator delay of a Flink job. The unit is **ms**.
Flink Job Maximum Operator Backpressure	Displays the maximum operator backpressure value of a Flink job. A larger value indicates severer backpressure. **0**: OK **50**: low **100**: high

----End

Viewing the Task List of a Job

You can view details about each task running on a job, including the task start time, number of received and transmitted bytes, and running duration.

NOTE

If the value is **0**, no data is received from the data source.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click the name of the job you want. The job details are displayed.

Step 3 On **Task List** and view the node information about the task.

View the operator task list. The following table describes the task parameters.

Table 5-13 Parameter description

Parameter	Description
Name	Name of an operator.
Duration	Running duration of an operator.
Max Concurrent Jobs	Number of parallel tasks in an operator.
Task	Operator tasks are categorized as follows: ● The digit in red indicates the number of failed tasks. ● The digit in light gray indicates the number of canceled tasks. ● The digit in yellow indicates the number of tasks that are being canceled. ● The digit in green indicates the number of finished tasks. ● The digit in blue indicates the number of running tasks. ● The digit in sky blue indicates the number of tasks that are being deployed. ● The digit in dark gray indicates the number of tasks in a queue.
Status	Status of an operator task.
Back Pressure Status	Working load status of an operator. Available options are as follows: ● **OK**: indicates that the operator is in normal working load. ● **LOW**: indicates that the operator is in slightly high working load. DLI processes data quickly. ● **HIGH**: indicates that the operator is in high working load. The data input speed at the source end is slow.
Delay	Duration from the time when source data starts being processed to the time when data reaches the current operator. The unit is millisecond.
Sent Records	Number of data records sent by an operator.
Sent Bytes	Number of bytes sent by an operator.
Received Bytes	Number of bytes received by an operator.
Received Records	Number of data records received by an operator.
Started	Time when an operator starts running.
Ended	Time when an operator stops running.

----End

Viewing the Job Execution Plan

You can view the execution plan to understand the operator stream information about the running job.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click the name of the job you want. The job details are displayed.

Step 3 Click the **Execution Plan** tab to view the operator flow direction.

Click a node. The corresponding information is displayed on the right of the page.

- Scroll the mouse wheel to zoom in or out.
- The stream diagram displays the operator stream information about the running job in real time.

----End

Viewing Job Submission Logs

You can view the submission logs to locate the fault.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click the name of the job you want. The job details are displayed.

Step 3 In the **Commit Logs** tab, view the information about the job submission process.

----End

Viewing Job Running Logs

You can view the run logs to locate the faults occurring during job running.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click the name of the job you want. The job details are displayed.

Step 3 On the **Run Log** tab page, you can view the **Job Manager** and **Task Manager** information of the running job.

Information about JobManager and TaskManager is updated every minute. Only run logs of the last minute are displayed by default.

If you select an OBS bucket for saving job logs during the job configuration, you can switch to the OBS bucket and download log files to view more historical logs.

If the job is not running, information on the **Task Manager** page cannot be viewed.

----End

5.3.9 Tag Management

A tag is a key-value pair customized by users and used to identify cloud resources. It helps users to classify and search for cloud resources. A tag consists of a tag key and a tag value.

DLI allows you to add tags to Flink jobs. You can add tags to Flink jobs to identify information such as the project name, service type, and background. If you use tags in other cloud services, you are advised to create the same tag key-value pairs for cloud resources used by the same business to keep consistency.

DLI supports the following two types of tags:

- Resource tags: indicate non-global tags created on DLI.
- Predefined tags: global tags created on Tag Management Service (TMS).

This section includes the following content:

- **Managing a Job Tag**
- **Searching for a Job by Tag**

Managing a Job Tag

DLI allows you to add, modify, or delete tags for jobs.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click the name of the job to be viewed. The **Job Details** page is displayed.

Step 3 Click **Tags** to display the tag information about the current job.

Step 4 Click **Add/Edit Tag** to open to the **Add/Edit Tag** dialog box.

Step 5 Configure the tag parameters in the **Add/Edit Tag** dialog box.

Table 5-14 Tag parameters

Parameter	Description
Tag key	You can perform the following operations: • Click the text box and select a predefined tag key from the drop-down list. To add a predefined tag, you need to create one on TMS and then select it from the **Tag key** drop-down list. You can click **View predefined tags** to go to the **Predefined Tags** page of the TMS console. Then, click **Create Tag** in the upper corner of the page to create a predefined tag. • Enter a tag key in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with **_sys_**.

Parameter	Description
Tag value	You can perform the following operations: • Click the text box and select a predefined tag value from the drop-down list. • Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 225 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

NOTE

- A maximum of 20 tags can be added.
- Only one tag value can be added to a tag key.
- The key name in each resource must be unique.

Step 6 Click **OK**.

Step 7 (Optional) In the tag list, locate the row where the tag you want to delete resides, click **Delete** in the **Operation** column to delete the tag.

----End

Searching for a Job by Tag

If tags have been added to a job, you can search for the job by setting tag filtering conditions to quickly find it.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 In the upper right corner of the page, click the search box and select **Tags**.

Step 3 Choose a tag key and value as prompted. If no tag key or value is available, create a tag for the job. For details, see **Managing a Job Tag**.

Step 4 Choose other tags to generate a tag combination for job search.

Step 5 Click search icon. The target job will be displayed in the job list.

----End

5.4 Spark Job Management

5.4.1 Spark Job Management

Based on the open-source Spark, DLI optimizes performance and reconstructs services to be compatible with the Apache Spark ecosystem and interfaces, and executes batch processing tasks.

DLI also allows you to use Spark jobs to access DLI metadata.

Spark job management provides the following functions:

- **Creating a Spark Job**
- **Re-executing a Job**
- **Searching for a Job**
- **Terminating a Job**

In addition, you can click **Quick Links** to switch to the details on *User Guide*.

Spark Jobs Page

On the **Overview** page, click **Spark Jobs** to go to the SQL job management page. Alternatively, you can click **Job Management** > **Spark Jobs**. The page displays all Spark jobs. If there are a large number of jobs, they will be displayed on multiple pages. DLI allows you to view jobs in all statuses.

Table 5-15 Job management parameters

Parameter	Description
Job ID	ID of a submitted Spark job, which is generated by the system by default.
Name	Name of a submitted Spark job.
Queues	Queue where the submitted Spark job runs
Username	Name of the user who executed the Spark job
Status	Job status. The following values are available: • **Starting**: The job is being started. • **Running**: The job is being executed. • **Failed**: The session has exited. • **Finished**: The session is successfully executed. • **Restoring**: The job is being restored.
Created	Time when a job is created. Jobs can be displayed in ascending or descending order of the job creation time.
Last Modified	Time when a job is completed.

Parameter	Description
Operation	• **Edit**: You can modify the current job configuration and re-execute the job. • **SparkUI**: After you click this button, the Spark job execution page is displayed. **NOTE** • The SparkUI page cannot be viewed for jobs in the **Starting** state. • Currently, only the latest 100 job information records are displayed on the SparkUI of DLI. • **Terminate Job**: Cancel a job that is being started or running. • **Re-execute**: Run the job again. • **Archive Log**: Save job logs to the temporary bucket created by DLI. • **Export Log**: Export logs to the created OBS bucket. **NOTE** • You have the permission to create OBS buckets. • If the job is in the **Running** state, logs cannot be exported. • **Commit Log**: View the logs of submitted jobs. • **Driver Log**: View the logs of running jobs.

Re-executing a Job

On the **Spark Jobs** page, click **Edit** in the **Operation** column of the job. On the Spark job creation page that is displayed, modify parameters as required and execute the job.

Searching for a Job

On the **Spark Jobs** page, select **Status** or **Queues**. The system displays the jobs that meet the filter condition in the job list.

Terminating a Job

On the **Spark Jobs** page, choose **More** > **Terminate Job** in the **Operation** column of the job that you want to stop.

Exporting Logs

On the **Spark Jobs** page, choose **More** > **Export Log** in the Operation column of the corresponding job. In the dialog box that is displayed, enter the path of the created OBS bucket and click **OK**.

5.4.2 Creating a Spark Job

DLI provides fully-managed Spark computing services by allowing you to execute Spark jobs.

On the **Overview** page, click **Create Job** in the upper right corner of the **Spark Jobs** tab or click **Create Job** in the upper right corner of the **Spark Jobs** page. The Spark job editing page is displayed.

On the Spark job editing page, a message is displayed, indicating that a temporary DLI data bucket will be created. The created bucket is used to store temporary data generated by DLI, such as job logs and job results. You cannot view job logs if you choose not to create it. The bucket will be created and the default bucket name is used.

If you do not need to create a DLI temporary data bucket and do not want to receive this message, select **Do not show again** and click **Cancel**.

Prerequisites

- You have uploaded the dependencies to the corresponding OBS bucket on the **Data Management > Package Management** page. For details, see **Creating a Package**.
- Before creating a Spark job to access other external data sources, such as OpenTSDB, HBase, Kafka, GaussDB(DWS), RDS, CSS, CloudTable, DCS Redis, and DDS MongoDB, you need to create a cross-source connection to enable the network between the job running queue and external data sources.
  - For details about the external data sources that can be accessed by Spark jobs, see **Cross-Source Analysis Development Methods**.
  - For details about how to create a datasource connection, see **Enhanced Datasource Connections**.

    On the **Resources** > **Queue Management** page, locate the queue you have created, and choose **More** > **Test Address Connectivity** in the **Operation** column to check whether the network connection between the queue and the data source is normal. For details, see **Testing Address Connectivity**.

Procedure

1. In the left navigation pane of the DLI management console, choose **Job Management** > **Spark Jobs**. The **Spark Jobs** page is displayed.

   Click **Create Job** in the upper right corner. In the job editing window, you can set parameters in **Fill Form** mode or **Write API** mode.

   The following uses the **Fill Form** as an example. In **Write API** mode, refer to the *Data Lake Insight API Reference* for parameter settings.

2. Select a queue.

   Select the queue you want to use from the drop-down list box.

3. Configure the job.

   Configure job parameters by referring to **Table 5-16**.

   Table 5-16 Job configuration parameters

Parameter	Description
Job Name (--name)	Set a job name.

Parameter	Description
Application	Select the package to be executed. The value can be **.jar** or **.py**.
Main Class (--class)	Enter the name of the main class. When the application type is **.jar**, the main class name cannot be empty.
Spark Arguments (--conf)	Enter a parameter in the format of **key=value**. Press Enter to separate multiple key-value pairs. These parameters can be replaced using global variables. For example, if you create a global variable **custom_class** on the **Global Configuration** > **Global Variables** page, you can use **"spark.sql.catalog"={{custom_class}}** to replace a parameter with this variable after the job is submitted. **NOTE** • The JVM garbage collection algorithm cannot be customized for Spark jobs. • If the Spark version is **3.1.1**, configure **Spark parameters (--conf)** to select a dependent module. For details about the configuration, see **Creating a Spark Job**.
Job Type	Set this parameter when you select a CCE queue. Type of the Spark image used by a job. The options are as follows: • **Basic**: Basic images provided by DLI. Select this option for non-AI jobs. • **AI-enhanced**: AI images provided by DLI. Select this option for AI jobs. • **Image**: Custom Spark images. Select an existing image name and version on SWR.
JAR Package Dependencies (--jars)	JAR file on which the Spark job depends. You can enter the JAR package name or the corresponding OBS path. The format is as follows: **obs://*Bucket name/Folder name/Package name***.
Python File Dependencies (--py-files)	py-files on which the Spark job depends. You can enter the Python file name or the corresponding OBS path of the Python file. The format is as follows: **obs://*Bucket name/Folder name/File name***.
Other Dependencies (--files)	Other files on which the Spark job depends. You can enter the name of the dependency file or the corresponding OBS path of the dependency file. The format is as follows: **obs://*Bucket name/Folder name/File name***.
Group Name	If you select a group when creating a package, you can select all the packages and files in the group. For details about how to create a package, see **Creating a Package**.
Access Metadata	Whether to access metadata through Spark jobs. .

Parameter	Description
Retry upon Failure	Indicates whether to retry a failed job. If you select **Yes**, you need to set the following parameters: **Maximum Retries**: Maximum number of retry times. The maximum value is **100**.
Advanced Settings	• Skip • Configure – **Select Dependency Resources**: For details about the parameters, see **Table 5-18**. – **Configure Resources**: For details about the parameters, see **Table 5-19**.

Table 5-17 Spark Parameter (--conf) configuration

Datasource	Example Value
CSS	spark.driver.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/css/* spark.executor.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/css/*
DWS	spark.driver.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/dws/* spark.executor.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/dws/*
HBase	spark.driver.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/hbase/* spark.executor.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/hbase/*
OpenTSDB	park.driver.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/opentsdb/* spark.executor.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/opentsdb/*
RDS	spark.driver.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/rds/* spark.executor.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/rds/*
Redis	spark.driver.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/redis/* spark.executor.extraClassPath=/usr/share/extension/dli/spark-jar/datasource/redis/*

4. Set the following parameters in advanced settings:

- **Select Dependency Resources**: For details about the parameters, see **Table 5-18**.
- **Configure Resources**: For details about the parameters, see **Table 5-19**.

Table 5-18 Parameters for selecting dependency resources

Parameter	Description
modules	If the Spark version is **3.1.1**, you do not need to select a module. Configure **Spark parameters (--conf)**. Dependency modules provided by DLI for executing datasource connection jobs. To access different services, you need to select different modules. • MRS HBase: sys.datasource.hbase • DDS: sys.datasource.mongo • MRS OpenTSDB: sys.datasource.opentsdb • DWS: sys.datasource.dws • RDS MySQL: sys.datasource.rds • RDS PostGre: sys.datasource.rds • DCS: sys.datasource.redis • CSS: sys.datasource.css
Resource Package	JAR package on which the Spark job depends.

Table 5-19 Resource specification parameters

Parameter	Description
Resource Specifications	Select a resource specification from the drop-down list box. The system provides three resource specifications for you to select. The following configuration items in the resource specifications can be modified: • Executor Memory • Executor Cores • Executors • Driver Cores • Driver Memory If modified, your modified settings of the items are used.
Executor Memory	Customize the configuration item based on the selected resource specifications.
Executor Cores	Customize the configuration item based on the selected resource specifications.

Parameter	Description
Executors	Customize the configuration item based on the selected resource specifications.
Driver Cores	Customize the configuration item based on the selected resource specifications.
Driver Memory	Customize the configuration item based on the selected resource specifications.

NOTE

Spark job parameter calculation:

- Number of CUs = Number of driver CPU cores + Number of executors x Number of executor CPU cores

  The cluster management plane and driver use some CU resources. Number of Executors * Number of Executor Cores must be smaller than the number of computing CUs of the queue.
- Memory = Driver memory + (Number of Executors x Executor memory)

5. Click **Execute** in the upper right corner of the Spark job editing page.

   After the message "Batch processing job submitted successfully" is displayed, you can view the status and logs of the submitted job on the **Spark Jobs** page.

6 Queue Management

6.1 Overview

Queue

Queues in DLI are computing resources, which are the basis for using DLI. All executed jobs require computing resources.

Currently, DLI provides two types of queues: **For SQL** and **For general purpose**.

- For SQL: The queue is used to run SQL jobs.
- For general purpose: The queue is used to run Spark programs, Flink SQL jobs, and Flink Jar jobs.

Constraints

- A queue named **default** is preset in DLI for you to experience. Resources are allocated on demand.
- Queue types:
  - For SQL: Spark SQL jobs can be submitted to SQL queues.
  - For general purpose: The queue is used to run Spark programs, Flink SQL jobs, and Flink Jar jobs.

  The queue type cannot be changed. If you want to use another queue type, purchase a new queue.
- The region of a queue cannot be changed.
- A newly created queue can be scaled in or out only after a job is executed on the queue.
- DLI queues cannot access the Internet.

Difference Between Computing and Storage Resources

Table 6-1 Difference between computing and storage resources

Resource	How to Obtain	Function
Compute resource	Create queue on the DLI management console.	Used for executing queries.
Storage resource	DLI has a 5 GB quota.	Used for storing data in the database and DLI tables.

NOTE

- Storage resources are internal storage resources of DLI for storing database and DLI tables and represent the amount of data stored in DLI.
- By default, DLI provides a 5 GB quota for storage resources.
- A queue named **default** is preset in DLI. If you are uncertain about the required queue capacity or have no available queue capacity to run queries, you can execute jobs using this queue.
- The **default** queue is used only for user experience. It may be occupied by multiple users at a time. Therefore, it is possible that you fail to obtain the resource for related operations. You are advised to use a self-built queue to execute jobs.

Dedicated Queue

Resources of a dedicated queue are not released when the queue is idle. That is, resources are reserved regardless of whether the queue is used. Dedicated queues ensure that resources exist when jobs are submitted.

Elastic Scaling

DLI allows you to flexibly scale in or out queues on demand. After a queue with specified specifications is created, you can scale it in and out as required.

To change the queue specifications, see **Elastic Scaling**.

NOTE

Scaling can be performed for a newly created queue only when jobs are running on this queue.

Scheduled Elastic Scaling

DLI allows you to schedule tasks for periodic queue scaling. After creating a queue, the scheduled scaling tasks can be executed.

You can schedule an automatic scale-out/scale-in based on service requirements. The system periodically triggers queue scaling. For details, see **Scheduling CU Changes**.

NOTE

Scaling can be performed for a newly created queue only when jobs are running on this queue.

Automatic Queue Scaling

Flink jobs use queues. DLI can automatically trigger scaling for jobs based on the job size.

NOTE

Scaling can be performed for a newly created queue only when there are jobs running on this queue.

Queue Management Page

Queue Management provides the following functions:

- **Managing Permissions**
- **Creating a Queue**
- **Deleting a Queue**
- **Modifying CIDR Block**
- **Elastic Scaling**
- **Scheduling CU Changes**
- **Testing Address Connectivity**
- **Creating a Topic for Key Event Notifications**

NOTE

To receive notifications when a DLI job fails, **SMN Administrator** permissions are required.

The queue list displays all queues created by you and the **default** queue. Queues are listed in chronological order by default in the queue list, with the most recently created queues displayed at the top.

Table 6-2 Parameter description

Parameter	Description
Name	Name of a queue.
Type	Queue type. • For SQL • For general purpose • Spark queue (compatible with earlier versions)
Specifications	Queue size. Unit: CU CU is the pricing unit of queues. A CU consists of 1 vCPU and 4-GB memory. The computing capabilities of queues vary with queue specifications. The higher the specifications, the stronger the computing capability.

Parameter	Description
Actual CUs	Actual size of the current queue.
Elastic Scaling	Target CU value for scheduled scaling, or the maximum and minimum CU values of the current specifications.
Username	Queue owner
Description	Description of a queue specified during queue creation. If no description is provided, **--** is displayed.
Operation	• **Delete**: Allow you to delete the selected queue. You cannot delete a queue where there are running jobs or jobs are being submitted. • **Manage Permissions**: You can view the user permissions corresponding to the queue and grant permissions to other users. • **More** – **Restart**: Forcibly restart a queue. **NOTE** Only the SQL queue has the **Restart** operation. – **Elastic Scaling**: You can select **Scale-out** or **Scale-in** as required. The number of CUs after modification must be an integer multiple of 16. – **Schedule CU Changes**: You can set different queue sizes at different time or in different periods based on the service period or usage. The system automatically performs scale-out or scale-in as scheduled. The **After Modification** value must be an integer multiple of 16. – **Modifying CIDR Block**: When DLI enhanced datasource connection is used, the CIDR block of the DLI queue cannot overlap with that of the data source. You can modify the CIDR block as required. – **Test Address Connectivity**: Test whether the queue is reachable to the specified address. The domain name and IP address are supported. The port can be specified.

6.2 Queue Permission Management

Scenario

- You can isolate queues allocated to different users by setting permissions to ensure data query performance.
- The administrator and queue owner have all permissions, which cannot be set or modified by other users.

Operations

Step 1 On the top menu bar of the DLI management console, click **Resources** > **Queue Management**.

Step 2 Select the queue to be configured and choose **Manage Permissions** in the **Operation** column. The **User Permission Info** area displays the list of users who have permissions on the queue.

You can assign queue permissions to new users, modify permissions for users who have some permissions of a queue, and revoke all permissions of a user on a queue.

- Assign permissions to a new user.

  A new user does not have permissions on the queue.

  a. Click **Set Permission** on the right of **User Permissions** page. The **Set Permission** dialog box is displayed.

  b. Specify **Username** and select corresponding permissions.

  c. Click **OK**.

  Table 6-3 describes the related parameters.

Table 6-3 Parameter description

Parameter	Description
Username	Name of the authorized user. **NOTE** The username is an existing IAM user name and has logged in to the DLI management console.

Parameter	Description
Permission Settings	▪ **Delete Queues**: This permission allows you to delete the queue. ▪ **Submit Jobs**: This permission allows you to submit jobs using this queue. ▪ **Terminate Jobs**: This permission allows you to terminate jobs submitted using this queue. ▪ **Grant Permission**: This permission allows you to grant queue permissions to other users. ▪ **Revoke Permission**: This permission allows you to revoke the queue permissions that other users have but cannot revoke the queue owner's permissions. ▪ **View Other User's Permissions**: This permission allows you to view the queue permissions of other users. ▪ **Restart Queues**: This permission allows you to restart queues. ▪ **Modify Queue Specifications**: This permission allows you to modify queue specifications.

- To assign or revoke permissions of a user who has some permissions on the queue, perform the following steps:
  a. In the list under **User Permission Info** for a queue, select the user whose permissions need to be modified and click **Set Permission** in the **Operation** column.
  b. In the displayed **Set Permission** dialog box, modify the permissions of the current user. **Table 6-3** lists the detailed permission descriptions.
     If all options under **Set Permission** are gray, you are not allowed to change permissions on this queue. You can apply to the administrator, queue owner, or other authorized users for queue permission granting and revoking.
  c. Click **OK**.
- To revoke all permission of a user on the queue, perform the following steps:
  In the user list under **Permission Info**, select the user whose permission needs to be revoked and click **Revoke Permission** under **Operation**. In the **Revoke Permission** dialog box, click **OK**. All permissions on this queue are revoked.

----End

6.3 Creating a Queue

Before executing a job, you need to create a queue.

NOTE

- If you use a sub-account to create a queue for the first time, log in to the DLI management console using the main account and keep records in the DLI database before creating a queue.
- It takes 6 to 10 minutes for a job running on a new queue for the first time.
- After a queue is created, if no job is run within one hour, the system releases the queue.

Procedure

1. You can create a queue on the **Overview**, **SQL Editor**, or **Queue Management** page.
   - In the upper right corner of the **Overview** page, click Create Queue.
   - To create a queue on the **Queue Management** page:
     i. In the navigation pane of the DLI management console, choose **Resources** >**Queue Management**.
     ii. In the upper right corner of the **Queue Management** page, click **Create Queue** to create a queue.
   - To create a queue on the **SQL Editor** page:
     i. In the navigation pane of the DLI management console, click **SQL Editor**.
     ii. Click **Queues**. On the tab page displayed, click  on the right to create a queue.
2. On the **Create Queue** page displayed, set the parameters according to **Table 6-4**.

Table 6-4 Parameters

Parameter	Description
Name	Name of a queue. • The queue name can contain only digits, letters, and underscores (_), but cannot contain only digits, start with an underscore (_), or be left unspecified. • The length of the name cannot exceed 128 characters. **NOTE** The queue name is case-insensitive. Uppercase letters will be automatically converted to lowercase letters.
Type	• **For SQL**: compute resources used for SQL jobs. • **For general purpose**: compute resources used for Spark and Flink jobs. **NOTE** Selecting **Dedicated Resource Mode** enables you to create a dedicated queue. Enhanced datasource connections can only be created for dedicated queues.

Parameter	Description
Specifications	The compute nodes' total number of CUs. One CU equals one vCPU and 4 GB of memory. DLI automatically assigns CPU and memory resources to each compute node, and the client does not need to know how many compute nodes are being used.
Description	Description of the queue to be created. The description can contain a maximum of 128 characters.
Advanced Settings	In the **Queue Type** area, select **Dedicated Resource Mode** and then click **Advanced Settings**. • **Default**: The system automatically configures the parameter. • **Custom** **CIDR Block**: You can specify the CIDR block. For details, see **Modifying the CIDR Block**. If DLI enhanced datasource connection is used, the CIDR block of the DLI queue cannot overlap with that of the data source. **Queue Type**: When running an AI-related SQL job, select **AI-enhanced**. When running other jobs, select **Basic**.
Tags	Tags used to identify cloud resources. A tag includes the tag key and tag value. If you want to use the same tag to identify multiple cloud resources, that is, to select the same tag from the drop-down list box for all services, you are advised to create predefined tags on the Tag Management Service (TMS). **NOTE** • A maximum of 20 tags can be added. • Only one tag value can be added to a tag key. • The key name in each resource must be unique. • Tag key: Enter a tag key name in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with **_sys_**. • Tag value: Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 225 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

3. Click **Create Now** to create a queue.

   After a queue is created, you can view and select the queue for use on the **Queue Management** page.

   NOTE

   It takes 6 to 10 minutes for a job running on a new queue for the first time.

6.4 Deleting a Queue

You can delete a queue based on actual conditions.

NOTE

- This operation will fail if there are jobs in the **Submitting** or **Running** state on this queue.
- Deleting a queue does not cause table data loss in your database.

Procedure

Step 1 On the left of the DLI management console, click **Resources** >**Queue Management**.

Step 2 Locate the row where the target queue locates and click **Delete** in the **Operation** column.

NOTE

If **Delete** in the **Operation** column is gray, the current user does not have the permission of deleting the queue. You can apply to the administrator for the permission.

Step 3 In the displayed dialog box, click **OK**.

----End

6.5 Modifying the CIDR Block

If the CIDR block of the DLI queue conflicts with that of the user data source, you can change the CIDR block of the queue.

NOTE

If the queue whose CIDR block is to be modified has jobs that are being submitted or running, or the queue has been bound to enhanced datasource connections, the CIDR block cannot be modified.

Procedure

1. On the left of the DLI management console, click **Resources** >**Queue Management**.
2. Select the queue to be modified and click **Modify CIDR Block** in the **Operation** column.
3. Enter the required CIDR block and click **OK**. After the CIDR block of the queue is successfully changed, wait for 5 to 10 minutes until the cluster to which the queue belongs is restarted and then run jobs on the queue.

6.6 Elastic Queue Scaling

Prerequisites

Elastic scaling can be performed for a newly created queue only when there were jobs running in this queue.

Precautions

- If **Status of queue xxx is assigning, which is not available** is displayed on the **Elastic Scaling** page, the queue can be scaled only after the queue resources are allocated.

Scaling Out

If the current queue specifications do not meet service requirements, you can add the number of CUs to scale out the queue.

NOTE

Scale-out is time-consuming. After you perform scale-out on the **Elastic Scaling** page of DLI, wait for about 10 minutes. The duration is related to the CU amount to add. After a period of time, refresh the **Queue Management** page and check whether values of **Specifications** and **Actual CUs** are the same to determine whether the scale-out is successful. Alternatively, on the **Job Management** page, check the status of the **SCALE_QUEUE** SQL job. If the job status is **Scaling**, the queue is being scaled out.

The procedure is as follows:

1. On the left of the DLI management console, click **Resources** > **Queue Management**.
2. Select the queue to be scaled out and choose **More > Elastic Scaling** in the **Operation** column.
3. On the displayed page, select **Scale-out** for **Operation** and set the scale-out amount.
4. Click .

Scaling In

If the current queue specifications are too much for your computing service, you can reduce the number of CUs to scale in the queue.

NOTE

- Scale-in is time-consuming. After you perform scale-in on the **Elastic Scaling** page of DLI, wait for about 10 minutes. The duration is related to the CU amount to reduce. After a period of time, refresh the **Queue Management** page and check whether values of **Specifications** and **Actual CUs** are the same to determine whether the scale-in is successful. Alternatively, on the **Job Management** page, check the status of the **SCALE_QUEUE** SQL job. If the job status is **Scaling**, the queue is being scaled in.
- The system may not fully scale in the queue to the target size. If the current queue is in use or the service volume of the queue is large, the scale-in may fail or only partial specifications may be reduced.
- By default, the minimum number of CUs is **16**. That is, when the queue specifications are **16 CUs**, you cannot scale in the queue.

The procedure is as follows:

1. On the left of the DLI management console, click **Resources** > **Queue Management**.
2. Select the queue to be scaled out, click **More** in the **Operation** column, and select **Elastic Scaling**.
3. On the displayed page, select **Scale-in** for **Operation** and set the scale-in amount.
4. Click .

6.7 Scheduling CU Changes

Scenario

When services are busy, you might need to use more compute resources to process services in a period. After this period, you do not require the same amount of resources. If the purchased queue specifications are small, resources may be insufficient during peak hours. If the queue specifications are large, resources may be wasted.

DLI provides scheduled tasks for elastic scale-in and -out in the preceding scenario. You can set different queue sizes (CUs) at different time or in different periods based on your service period or usage and the existing queue specifications to meet your service requirements and reduce costs.

Precautions

- Periodic scaling can be performed for a newly created queue only when there were jobs running in this queue.
- Scheduled scaling tasks are available only for a queue with more than 64 CUs. That is, the minimum specifications of a queue are 64 CUs.
- A maximum of 12 scheduled tasks can be created for each queue.
- When each scheduled task starts, the actual start time of the specification change has a deviation of 5 minutes. It is recommended that the task start time be at least 20 minutes earlier than the time when the queue is actually used.
- The interval between two scheduled tasks must be at least 2 hours.

- Changing the specifications of a queue is time-consuming. The time required for changing the specifications depends on the difference between the target specifications and the current specifications. You can view the specifications of the current queue on the **Queue Management** page.
- If a job is running in the current queue, the queue may fail to be scaled in to the target CU amount value. Instead, it will be scaled in to a value between the current queue specifications and the target specifications. The system will try to scale in again 1 hour later until the next scheduled task starts.
- If a scheduled task does not scale out or scale in to the target CU amount value, the system triggers the scaling plan again 15 minutes later until the next scheduled task starts.

Creating Periodic Task

- If only scale-out or scale-in is required, you need to create only one task for changing specifications. Set the **Task Name**, **Final CU Count**, and **Executed** parameters. For details, see **Table 6-5**.
- To set both scale-out and scale-in parameters, you need to create two periodic tasks, and set the **Task Name**, **Final CU Count**, and **Executed** parameters. For details, see **Table 6-5**.

The procedure is as follows:

1. On the left of the DLI management console, click **Resources** > **Queue Management**.
2. Locate the queue for which you want to schedule a periodic task for elastic scaling, and choose **More** > **Schedule CU Changes** in the **Operation** column.
3. On the displayed page, click **Create Periodic Task** in the upper right corner.
4. On the **Create Periodic Task** page, set the required parameters. Click **OK**.

Table 6-5 Parameter description

Parameter	Description
Task Name	Enter the name of the periodic task. • The task name can contain only digits, letters, and underscores (_), but cannot contain only digits or start with an underscore (_) or be left unspecified. • The name can contain a maximum of 128 characters.
Enable Task	Whether to enable periodic elastic scaling. The task is enabled by default. If disabled, the task will not be triggered on time.
Validity Period	Time segment for executing the periodic task. The options include **Date** and **Time**.
Actual CUs	Queue specifications before scale-in or scale-out.

Parameter	Description
Final CUs	Specifications after the queue is scaled in or out. **NOTE** • By default, the maximum specifications of a queue are **512 CUs**. • The minimum queue specifications for scheduled scaling are 64 CUs. That is, only when **Actual CUs** are more than 64 CUs, the scheduled scaling can be performed. • The value of **Actual CUs** must be a multiple of 16.
Repeat	Time when scheduled scale-out or scale-in is repeat. Scheduled tasks can be scheduled by week in **Repeat**. • By default, this parameter is not configured, indicating that the task is executed only once at the time specified by **Executed**. • If you select all, the plan is executed every day. • If you select some options of **Repeat**, the plan is executed once a week at all specified days. **NOTE** • You do not need to set this parameter if you only need to perform scale-in or scale-out once. • If you have set scaling, you can set **Repeat** as required. You can also set the repeat period together with the validity period.
Executed	Time when scheduled scale-out or scale-in is performed • When each scheduled task starts, the actual start time of the specification change has a deviation of 5 minutes. It is recommended that the task start time be at least 20 minutes earlier than the time when the queue is actually used. • The interval between two scheduled tasks must be at least 2 hours.

After a periodic task is created, you can view the specification change of the current queue and the latest execution time on the page for scheduling CU changes.

Alternatively, on the **Queue Management** page, check whether the **Specifications** change to determine whether the scaling is successful.

You can also go to the **Job Management** page and check the status of the **SCALE_QUEUE** job. If the job status is **Scaling**, the queue is being scaled in or out.

Modifying a Scheduled Task

If a periodic task cannot meet service requirements anymore, you can modify it on the **Schedule CU Changes** page.

1. In the navigation pane of the DLI management console, choose **Resources** >**Queue Management**.

2. Locate the queue for which you want to schedule a periodic task for elastic scaling, and choose **More** > **Schedule CU Changes** in the **Operation** column.
3. On the displayed page, click **Modify** in the **Operation** column. In the displayed dialog box, modify the task parameters as needed.

Deleting a Scheduled Task

If you do not need the task anymore, delete the task on the **Schedule CU Changes** page.

1. In the navigation pane of the DLI management console, choose **Resources** >**Queue Management**.
2. Locate the queue for which you want to schedule a periodic task for elastic scaling, and choose **More** > **Schedule CU Changes** in the **Operation** column.
3. On the displayed page, click **Delete** in the **Operation** column. In the displayed dialog box, click **OK**.

6.8 Testing Address Connectivity

It can be used to test the connectivity between the DLI queue and the peer IP address specified by the user in common scenarios, or the connectivity between the DLI queue and the peer IP address bound to the datasource connection in cross-source connection scenarios. The operation is as follows:

1. On the **Queue Management** page, locate the row containing the target queue, click **More** in the **Operation** column, and select **Test Address Connectivity**.
2. On the **Test Address Connectivity** page, enter the address to be tested. The domain name and IP address are supported, and the port number can be specified.
3. Click **Test**.

   If the test address is reachable, a message is displayed on the page, indicating that the address is reachable.

   If the test address is unreachable, the system displays a message indicating that the address is unreachable. Check the network configurations and try again. Network configurations include the VPC peering and the datasource connection. Check whether they have been activated.

6.9 Creating an SMN Topic

Scenario

Once you have created an SMN topic, you can easily subscribe to it by going to the **Topic Management** > **Topics** page of the SMN console. You can choose to receive notifications via SMS or email. After the subscription is successful, if a job fails, the system automatically sends a message to the subscription endpoint you specified.

Procedure

1. On the **Resources** > **Queue Management** page, click **Create SMN Topic** on the upper left side. The **Create SMN Topic** dialog box is displayed.
2. Select a queue and click **OK**.

   NOTE

   - You can select a single queue or all queues.
   - If you create a topic for a queue and another topic for all queues, the SMN of all queues does not include the message of the single queue.
   - After a message notification topic is created, you will receive a message notification only when a Spark job created on the subscription queue fails.

3. Click **Topic Management** in to go to the **Topic Management** page of the SMN service.
4. In the **Operation** column of the topic, click **Add Subscription**. Select **Protocol** to determine the subscription mode.
5. After you click the link in the email, you will receive a message indicating that the subscription is successful.
6. Go to the **Subscriptions** page of SMN, and check that subscription status is **Confirmed**.

6.10 Managing Queue Tags

Tag Management

A tag is a key-value pair that you can customize to identify cloud resources. It helps you to classify and search for cloud resources. A tag consists of a tag key and a tag value.

If you use tags in other cloud services, you are advised to create the same tag (key-value pairs) for cloud resources used by the same business to keep consistency.

DLI supports the following two types of tags:

- Resource tags: non-global tags created on DLI
- Predefined tags: global tags created on Tag Management Service (TMS).

DLI allows you to add, modify, or delete tags for queues.

Step 1 In the navigation pane of the DLI management console, choose **Resources** > **Queue Management**.

Step 2 In the **Operation** column of the queue, choose **More** > **Tags**.

Step 3 The tag management page is displayed, showing the tag information about the current queue.

Step 4 Click **Add/Edit Tag** to switch to the **Add/Edit Tag** dialog box. Enter a tag and a value, and click **Add**.

Table 6-6 Tag parameters

Parameter	Description
Tag key	You can specify the tag key in either of the following ways: ● Click the text box and select a predefined tag key from the drop-down list. To add a predefined tag, you need to create one on TMS and then select it from the **Tag key** drop-down list. You can click **View predefined tags** to go to the **Predefined Tags** page of the TMS console. Then, click **Create Tag** in the upper corner of the page to create a predefined tag. ● Enter a tag key in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with **_sys_**.
Tag value	You can specify the tag value in either of the following ways: ● Click the text box and select a predefined tag value from the drop-down list. ● Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 225 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

NOTE

- A maximum of 20 tags can be added.
- Only one tag value can be added to a tag key.
- The key name in each resource must be unique.

Step 5 Click **OK**.

Step 6 (Optional) To delete a tag, locate the row where the tag resides in the tag list and click **Delete** in the **Operation** column to delete the tag.

----End

7 Data Management

7.1 Databases and Tables

7.1.1 Overview

DLI database and table management provide the following functions:

- **Database Permission Management**
- **Table Permission Management**
- **Creating a Database or a Table**
- **Deleting a Database or a Table**
- **Changing the Owners of Databases and Tables**
- **Importing Data**
- **Exporting Data**
- **Viewing Metadata**
- **Previewing Data**

Difference Between DLI Tables and OBS Tables

- Data stored in DLI tables is applicable to delay-sensitive services, such as interactive queries.
- Data stored in OBS tables is applicable to delay-insensitive services, such as historical data statistics and analysis.

Constraints

- **Database**
  - **default** is the database built in DLI. You cannot create a database named **default**.
  - DLI supports a maximum of 50 databases.
- **Table**
  - DLI supports a maximum of 5,000 tables.

  - DLI supports the following table types:
    - **MANAGED**: Data is stored in a DLI table.
    - **EXTERNAL**: Data is stored in an OBS table.
    - **View**: A view can only be created using SQL statements.
    - Datasource table: The table type is also **EXTERNAL**.
  - You cannot specify a storage path when creating a DLI table.
- **Data import**
  - Only OBS data can be imported to DLI or OBS.
  - You can import data in CSV, Parquet, ORC, JSON, or Avro format from OBS to tables created on DLI.
  - To import data in CSV format to a partitioned table, place the partition column in the last column of the data source.
  - The encoding format of imported data can only be UTF-8.
- **Data export**
  - Data in DLI tables (whose table type is **MANAGED**) can only be exported to OBS buckets, and the export path must contain a folder.
  - The exported file is in JSON format, and the text format can only be UTF-8.
  - Data can be exported across accounts. That is, after account B authorizes account A, account A has the permission to read the metadata and permission information of account B's OBS bucket as well as the read and write permissions on the path. Account A can export data to the OBS path of account B.

Databases and Tables Page

The **Databases and Tables** page displays all created databases. You can view the database information, such as the owner and the number of tables.

Table 7-1 Database and table management parameters

Parameter	Description
Database Name	• The database name can contain only digits, letters, and underscores (_), but cannot contain only digits or start with an underscore (_). • The database name is case insensitive and cannot be left unspecified. • It cannot exceed 128 characters.
Username	Database owner.
Tables	Number of tables in the database.
Description	Description of the database specified during database creation. If no description is provided, -- is displayed.

Parameter	Description
Enterprise Project	Enterprise project to which the database belongs. An enterprise project facilitates project-level management and grouping of cloud resources and users.
Operation	• **Permissions**: View the permission information and perform user authorization, permission settings, and user permission revocation. • **Tables**: View the tables in the corresponding database. For details, see **Table Management Page**. • **Create Table**: This permission allows you to create a table in the corresponding database. • **Modify Database**. This permission allows you to change the owner of the database. The username must exist under the same account. • **Drop Database**: This permission allows you to delete the selected database.

Table Management Page

From the **Data Management** page, click the database name or **Tables** in the **Operation** column to switch to the table management page.

The displayed page lists all tables created in the current database. You can view the table type, data storage location, and other information. Tables are listed in chronological order by default, with the most recently created tables displayed at the top.

Table 7-2 Table management parameters

Parameter	Description
Table Name	• The table name can contain only digits, letters, and underscores (_), but cannot contain only digits or start with an underscore (_). • The table name is case insensitive and cannot be left unspecified. • The table name can contain the dollar sign ($). An example value is **$test**. • It cannot exceed 128 characters.

Parameter	Description
Table Type	Table type. Available options are as follows: • **Managed**: Indicates that data is stored in a DLI table. • **External**: Indicates that data is stored in an OBS table. • **View**: Indicates the view type. You can only create views using SQL statements. **NOTE** The table or view information contained in the view cannot be modified. If the table or view information is modified, the query may fail.
Owner	User who creates the table.
Storage Location	DLI, OBS, View, CloudTable, and CSS data location
Size	Size of the data in the table. The value is displayed only for tables of the **Managed** type. For tables of other types, **--** is displayed.
Data Source Path	• If **Data Location** is **OBS**, the corresponding OBS path is displayed. • If **Data Location** is **DLI** and **View**, **--** is displayed. • When the data storage location is a datasource connection service such as CloudTable and CSS, the corresponding URL is displayed.
Created	Time when the table is created.
Last Accessed	Last time when an operation was performed on the table.
Operation	• **Manage Permissions**: This operation allows you to view the permission information and perform user authorization, permission settings, and user permission revocation. • **More**: – **Delete**: Delete a table from the corresponding database. – **Modify Owner**: Change the owner of a table The username must exist under the same account. – **Import**: Import data stored in an OBS bucket to a DLI or OBS table. – **Properties**: View data in **Metadata** and **Preview** tabs.

7.1.2 Managing Database Permissions

Scenario

- You can isolate databases allocated to different users by setting permissions to ensure data query performance.

- The administrator and database owner have all permissions, which cannot be set or modified by other users.

Precautions

- Lower-level objects automatically inherit permissions granted to upper-level objects. The hierarchical relationship is database > table > column.
- The database owner, table owner, and **authorized** users can assign permissions on the database and tables.
- Columns can only inherit the query permission. For details about **Inheritable Permissions**, see **Managing Database Permissions**.
- The permissions can be revoked only at the initial level to which the permissions are granted. You need to grant and revoke permissions at the same level. You need to grant and revoke permissions at the same level. For example, after you are granted the insertion permission on a database, you can obtain the insertion permission on the tables in the database. Your insertion permission can be revoked only at the database level.
- If you create a database with the same name as a deleted database, the database permissions will not be inherited. In this case, you need to grant the database permissions to users or projects.

  For example, user A is granted with the permission to delete the **testdb** database. Delete the database and create another one with the same name. You need to grant user A the deletion permission of the **testdb** database again.

Viewing Database Permissions

1. On the left of the management console, choose **Data Management** > **Databases and Tables**.
2. Locate the row where the target database resides and click **Manage Permissions** in the **Operation** column.

   Permissions can be granted to new users or projects, modified for users or projects with existing permissions, or revoked from a user or project.

Granting Permissions to a New User or Project

Here, the new user or project refers to a user or a project that does not have permissions on the database.

1. Click a database you need. In the displayed **Database Permission Management** page, click **Grant Permission** in the upper right corner.
2. In the displayed dialog box, select **User** or **Project**, enter the username or select the project to be authorized, and select the required permissions. For details about the permissions, see **Table 7-3**.

Table 7-3 Parameters

Parameter	Description
Authorization Object	Select **User** or **Project**.

Parameter	Description
Username/ Project Name	• If you select **User**, enter the IAM username when adding a user to the database. **NOTE** The username is an existing IAM user name and has logged in to the DLI management console. • If you select **Project**, select the project to be authorized in the current region. **NOTE** When **Project** is selected: • If you set **Non-inheritable Permissions**, you cannot view tables in the corresponding database within the project. • If you set **Inheritable Permissions**, you can view all tables in the database within the project.

Parameter	Description
Non-Inherited Permissions	Select a permission to grant it to the user, or deselect a permission to revoke it. Non-inherited permissions apply only to the current database. • The following permissions are applicable to both user and project authorization: – **Drop Database**: This permission allows you to delete the current database. – **Create Table**: This permission allows you to create tables in the current database. – **Create View**: This permission allows you to create views in the current database. – **Execute SQL EXPLAIN**: This permission allows you to execute an EXPLAIN statement and view information about how this database executes a query. – **Create Role**: This permission allows you to create roles in the current database. – **Delete Role**: This permission allows you to delete roles of the current database. – **View Role**: This permission allows you to view the role of the current user. – **Bind Role**: This permission allows you to bind roles to the current database. – **Unbind Role**: This permission allows you to bind roles from the current database. – **View All Binding Relationships**: This permission allows you to view the binding relationships between all roles and users. – **Create Function**: This permission allows you to create a function in the current database. – **Delete Function**: This permission allows you to delete functions from the current database. – **View All Functions**: This permission allows you to view all functions in the current database. – **View Function Details**: This permission allows you to view details about the current function. • The following permissions can only be granted to users: – **View All Tables**: This permission allows you to view all tables in the current database. **NOTE** If this permission of a specific database is not granted, all tables in the database will not be displayed. – **View Database**: This permission allows you to view the information about the current database.

Parameter	Description
	NOTE If this permission is not granted, the database will not be displayed.

Parameter	Description
Inherited Permissions	Select a permission to grant it to the user, or deselect a permission to revoke it. Inherited permissions are applicable to the current database and all its tables. However, only the query permission is applicable to table columns. The following permissions can be granted to both user and project. • **Drop Table**: This permission allows you to delete tables in a database. • **Select Table**: This permission allows you to query data of the current table. • **View Table Information**: This permission allows you to view information about the current table. • **Insert**: This permission allows you to insert data into the current table. • **Add Column**: This permission allows you to add columns to the current table. • **Overwrite**: This permission allows you to insert data to overwrite the data in the current table. • **Grant Permission**: This permission allows you to grant database permissions to other users or projects. • **Revoke Permission**: This permission allows you to revoke the permissions of the database that other users have but cannot revoke the database owner's permissions. • **Add Partition to Partition Table**: This permission allows you to add a partition to a partition table. • **Delete Partition from Partition Table**: This permission allows you to delete existing partitions from a partition table. • **Configure Path for Partition**: This permission allows you to set the path of a partition in a partition table to a specified OBS path. • **Rename Table Partition**: This permission allows you to rename partitions in a partition table. • **Rename Table**: This permission allows you to rename tables. • **Restore Table Partition**: This permission allows you to export partition information from the file system and save the information to metadata. • **View All Partitions**: This permission allows you to view all partitions in a partition table. • **View Other Users' Permissions**: This permission allows you to query other users' permission on the current database.

3. Click **OK**.

Modifying Permissions for an Existing User or Project

For a user or project that has some permissions on the database, you can revoke the existing permissions or grant new ones.

NOTE

If the options in **Set Permission** are gray, the corresponding account does not have the permission to modify the database. You can apply to the administrator, database owner, or other authorized users for granting and revoking permissions of databases.

1. In the **User Permission Info** list, find the user whose permission needs to be set.
   - If the user is a sub-user, you can set permissions for it.
   - If the user is already an administrator, you can only view the permissions information.

   In the **Project Permission Info** list, locate the project for which you want to set permissions and click **Set Permission**.
2. In the **Operation** column of the sub-user or project, click **Set Permission**. The **Set Permission** dialog box is displayed.

   For details about the permissions of database users or projects, see **Table 7-3**.
3. Click **OK**.

Revoking All Permissions of a User or Project

Revoke all permissions of a user or a project.

- In the user list under **User Permission Info**, locate the row where the target sub-user resides and click **Revoke Permission** in the **Operation** column. In the displayed dialog box, click **OK**. In this case, the user has no permissions on the database.

  NOTE

  If a user is an administrator, **Revoke Permission** is gray, indicating that the user's permission cannot be revoked.
- In the **Project Permission Info** area, select the project whose permissions need to be revoked and click **Revoke Permission** in the **Operation** column. After you click **OK**, the project does not have any permissions on the database.

7.1.3 Managing Table Permissions

Operation Scenario

- You can isolate databases allocated to different users by setting permissions to ensure data query performance.
- The administrator and database owner have all permissions, which cannot be set or modified by other users.

- When setting database permissions for a new user, ensure that the user group to which the user belongs has the **Tenant Guest** permission.

Precautions

- If you create a table with the same name as a deleted table, the table permissions will not be inherited. In this case, you need to grant the table permissions to users or projects.

  For example, user A is granted with the permission to delete the **testTable** table. Delete the table and create another one with the same name. You need to grant user A the deletion permission of the **testTable** table again.

Viewing Table Permissions

1. On the left of the management console, choose **Data Management** > **Databases and Tables**.
2. Click the database name in the table whose authority is to be set. The **Table Management** page of the database is displayed.
3. Locate the row where the target table resides and click **Manage Permissions** in the **Operation** column.

   Permissions can be granted to new users or projects, modified for users or projects with existing permissions, or revoked from a user or project.

Granting Permissions to a New User or a Project

Here, the new user or project refers to a user or a project that does not have permissions on the database.

1. Click the table you need. In the displayed table permissions page, click **Grant Permission** in the upper right corner.
2. In the displayed **Grant Permission** dialog box, select the required permissions.
   - For details about the DLI table permissions, see **Table 7-4**.

Table 7-4 Parameter description

Parameter	Description
Authorization Object	Select **User** or **Project**.
Username/ Project	• If you select **User**, enter the IAM username when granting table permissions to the user. **NOTE** The username is an existing IAM user name and has logged in to the DLI management console. • If you select **Project**, select the project to be authorized in the current region. **NOTE** If you select **Project**, you can only view information about the authorized tables and their databases.

Parameter	Description
Non-inheritable Permissions	Select a permission to grant it to the user, or deselect a permission to revoke it. • The following permissions are applicable to both user and project authorization: – **Select Table**: This permission allows you to query data of the current table. – **View Table Information**: This permission allows you to view information about the current table. – **View Table Creation Statement**: This permission allows you to view the statement for creating the current table. – **Drop Table**: This permission allows you to delete the current table. – **Rename Table**: Rename the current table. – **Insert**: This permission allows you to insert data into the current table. – **Overwrite**: This permission allows you to insert data to overwrite the data in the current table. – **Add Column**: Add columns to the current table. – **Grant Permission**: The current user can grant table permissions to other users. – **Revoke Permission**: The current user can revoke the table's permissions that other users have but cannot revoke the table owner's permissions. – **View Other Users' Permissions**: This permission allows you to query other users' permission on the current table. The partition table also has the following permissions: – **Delete Partition**: This permission allows you to delete existing partitions from a partition table. – **View All Partitions**: This permission allows you to view all partitions in a partition table. • The following permissions can only be granted to users: – **View Table**: This permission allows you to display the current table.

– For details about the OBS table permissions, see **Table 7-5**.

Table 7-5 Parameter description

Parameter	Description
Authorization Object	Select **User** or **Project**.
Username/ Project	• If you select **User**, enter the IAM username when granting table permissions to the user. **NOTE** The username is an existing IAM user name and has logged in to the DLI management console. • If you select **Project**, select the project to be authorized in the current region. **NOTE** If you select **Project**, you can only view information about the authorized tables and their databases.

Parameter	Description
Non-inheritable Permissions	Select a permission to grant it to the user, or deselect a permission to revoke it. • The following permissions are applicable to both user and project authorization: – **View Table Creation Statement**: This permission allows you to view the statement for creating the current table. – **View Table Information**: This permission allows you to view information about the current table. – **Select Table**: This permission allows you to query data of the current table. – **Drop Table**: This permission allows you to delete the current table. – **Rename Table**: Rename the current table. – **Insert**: This permission allows you to insert data into the current table. – **Overwrite**: This permission allows you to insert data to overwrite the data in the current table. – **Add Column**: This permission allows you to add columns to the current table. – **Grant Permission**: This permission allows you to grant table permissions to other users or projects. – **Revoke Permission**: This permission allows you to revoke the table's permissions that other users or projects have but cannot revoke the table owner's permissions. – **View Other Users' Permissions**: This permission allows you to query other users' permission on the current table. The partition table also has the following permissions: – **Add Partition**: This permission allows you to add a partition to a partition table. – **Delete Partition**: This permission allows you to delete existing partitions from a partition table. – **Configure Path for Partition**: This permission allows you to set the path of a partition in a partition table to a specified OBS path. – **Rename Table Partition**: This permission allows you to rename partitions in a partition table. – **Restore Table Partition**: This permission allows you to export partition information from the file system and save the information to metadata. – **View All Partitions**: This permission allows you to view all partitions in a partition table.

Parameter	Description
	• The following permissions can only be granted to users: – **View Table**: This permission allows you to view the current table.

– For details about the view permissions, see **Table 7-6**.

NOTE

A view can be created only by using SQL statements. You cannot create a view on the **Create Table** page.

Table 7-6 Parameter description

Parameter	Description
Authorization Object	Select **User** or **Project**.
Username/ Project	• If you select **User**, enter the IAM username when adding a user to the database. **NOTE** The username is an existing IAM user name and has logged in to the DLI management console. • If you select **Project**, select the project to be authorized in the current region. **NOTE** If you select **Project**, you can only view information about the authorized tables and their databases.

Parameter	Description
Non-inheritable Permissions	Select a permission to grant it to the user, or deselect a permission to revoke it. • The following permissions are applicable to both user and project authorization: – **View Table Information**: This permission allows you to view information about the current table. – **View Table Creation Statement**: This permission allows you to view the statement for creating the current table. – **Drop Table**: This permission allows you to delete the current table. – **Select Table**: This permission allows you to query data of the current table. – **Rename Table**: Rename the current table. – **Grant Permission**: The current user or project can grant table permissions to other users or projects. – **Revoke Permission**: The current user or project can revoke the table's permissions that other users or projects have but cannot revoke the table owner's permissions. – **View Other Users' Permissions**: This permission allows you to query other users' permission on the current table. • Only applicable to – **View Table**: This permission allows you to view the current table.

3. Click **OK**.

Modifying Permissions for an Existing User or Project

For a user or project that has some permissions on the database, you can revoke the existing permissions or grant new ones.

NOTE

If all options under **Set Permission** are gray, you are not allowed to change permissions on this table. You can apply to the administrator, table owner, or other authorized users for granting and revoking table permissions.

1. In the **User Permission Info** list, find the user whose permission needs to be set.
   – If the user is a sub-user and is not the owner of the table, you can set permissions.
   – If the user is an administrator or table owner, you can only view permissions.

In the **Project Permission Info** list, locate the project for which you want to set permissions and click **Set Permission**.

2. In the **Operation** column of the sub-user or project, click **Set Permission**. The **Set Permission** dialog box is displayed.
   - For details about DLI table user or project permissions, see **Table 7-4**.
   - For details about OBS table user or project permissions, see **Table 7-5**.
   - For details about View table user or project permissions, see **Table 7-6**.
3. Click **OK**.

Revoking All Permissions of a User or Project

Revoke all permissions of a user or a project.

- In the user list under **User Permission Info**, locate the row where the target sub-user resides and click **Revoke Permission** in the **Operation** column. In the displayed dialog box, click **OK**. In this case, the user has no permissions on the table.

  NOTE

  In the following cases, **Revoke Permission** is gray, indicating that the permission of the user cannot be revoked.
  - The user is an administrator.
  - The sub-user is the owner of the table.
  - The sub-user has only inheritable permissions.
- In the **Project Permission Info** area, select the project whose permissions need to be revoked and click **Revoke Permission** in the **Operation** column. After you click **OK**, the project does not have any permissions on the table.

  NOTE

  If a project has only inheritable permissions, **Revoke Permission** is gray, indicating that the permissions of the project cannot be revoked.

7.1.4 Creating a Database or a Table

Definition of Database and Table in DLI

A database, built on the computer storage device, is a data warehouse where data is organized, stored, and managed based on its structure.

The table is an important part of the database. It consists of rows and columns. Each column functions as a field. Each value in a field (column) represents a type of data.

The database is a framework and the table contains data content. A database has one or more tables.

You can create databases and tables on the management console or using SQL statements. This section describes how to create a database and a table on the management console.

NOTE

A view can be created only by using SQL statements. You cannot create a view on the **Create Table** page.

Precautions

- If a folder and a file have the same name in the OBS directory, the file path is preferred as the path of the OBS table to be created.

Creating a Database

Step 1 You can create a database on either the **Data Management** page or the **SQL Editor** page.

- To create a database on the **Data Management** page:
  a. On the left of the management console, choose **Data Management** > **Databases and Tables**.
  b. In the upper right corner of the **Databases and Tables** page, click **Create Database** to create a database.
- To create a database on the **SQL Editor** page:
  a. On the left of the management console, click **SQL Editor**.
  b. In the navigation pane on the left, click  beside **Databases**.

Step 2 In the displayed **Create Database** dialog box, specify **Name** and **Description** by referring to **Table 7-7**.

Table 7-7 Description

Parameter	Description
Database Name	• The database name can contain only digits, letters, and underscores (_), but cannot contain only digits or start with an underscore (_). • The database name is case insensitive and cannot be left blank. • The length of the database name cannot exceed 128 characters. **NOTE** The **default** database is a built-in database. You cannot create the **default**. database.
Description	Description of a database.

Parameter	Description
Tag	Tags used to identify cloud resources. A tag includes the tag key and tag value. If you want to use the same tag to identify multiple cloud resources, that is, to select the same tag from the drop-down list box for all services, you are advised to create predefined tags on the Tag Management Service (TMS). **NOTE** • A maximum of 20 tags can be added. • Only one tag value can be added to a tag key. • The key name in each resource must be unique. • Tag key: Enter a tag key name in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with _**sys**_. • Tag value: Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 225 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

Step 3 Click **OK**.

After a database is created, you can view and select the database for use on the **Databases and Tables** page or **SQL Editor** page.

----End

Creating a Table

Before creating a table, ensure that a database has been created.

Step 1 You can create a table on either the **Databases and Tables** page or the **SQL Editor** page.

NOTE

Datasource connection tables, such as View tables, HBase (MRS) tables, OpenTSDB (MRS) tables, DWS tables, RDS tables, and CSS tables, cannot be created. You can use SQL to create views and datasource connection tables. For details, see sections **Creating a View** and **Creating a Datasource Connection Table** in the *Data Lake Insight SQL Syntax Reference*.

- To create a table on the **Data Management** page:
  a. On the left of the management console, choose **Data Management** > **Databases and Tables**.
  b. On the **Databases and Tables** page, select the database for which you want to create a table. In the **Operation** column, click **More** > **Create Table** to create a table in the current database.
- To create a table on the **SQL Editor** page:

a. On the left of the management console, click **SQL Editor**.

b. In the navigation pane of the displayed **SQL Editor** page, click **Databases**. You can create a table in either of the following ways:

   - Click a database name. In the **Tables** area, click  on the right to create a table in the current database.
   - Click  on the right of the database and choose **Create Table** from the shortcut menu to create a table in the current database.

Step 2 In the displayed **Create Table** dialog box, set parameters as required.

- If you set **Data Location** to **DLI**, set related parameters by referring to **Table 7-8**.
- If you set **Data Location** to **OBS**, set related parameters by referring to **Table 7-8** and **Table 7-9**.

Table 7-8 Common parameters

Parameter	Description	Example
Table Name	– The table name can contain only digits, letters, and underscores (_), but cannot contain only digits or start with an underscore (_). – The table name is case insensitive and cannot be left unspecified. – The table name can contain the dollar sign ($). An example value is **$test**. – The length of the table name cannot exceed 128 characters.	table01
Data Location	Data storage location. Currently, DLI and OBS are supported.	DLI
Description	Description of the table.	-
Column Type	Available values: **Normal** or **Partition**	Normal
Column	Name of a column in a table. The column name must contain at least one letter and can contain underscores (_). It cannot contain only digits. You can select **Normal** or **Partition**. Partition columns are dedicated to partition tables. User data is partitioned to improve query efficiency. **NOTE** The column name is case-insensitive and must be unique.	name

Parameter	Description	Example
Type	Data type of a column. This parameter corresponds to **Column Name**. – **string**: The data is of the string type. – **int**: Each integer is stored on four bytes. – **date**: The value ranges from 0000-01-01 to 9999-12-31. – **double**: Each number is stored on eight bytes. – **boolean**: Each value is stored on one byte. – **decimal**: The valid bits are positive integers between 1 to 38, including 1 and 38. The decimal digits are integers less than 10. – **smallint/short**: The number is stored on two bytes. – **bigint/long**: The number is stored on eight bytes. – **timestamp**: The data indicates a date and time. The value can be accurate to six decimal points. – **float**: Each number is stored on four bytes. – **tinyint**: Each number is stored on one byte. Only OBS tables support this data type.	string
Column Description	Description of a column.	-
Operation	– Add Column – Delete **NOTE** If the table to be created includes a great number of columns, you are advised to use SQL statements to create the table or import column information from the local EXCEL file.	-

Table 7-9 Parameter description when **Data Location** is set to **OBS**

Parameter	Description	Example
Data Format	DLI supports the following data formats: – **Parquet**: DLI can read non-compressed data or data that is compressed using Snappy and gzip. – **CSV**: DLI can read non-compressed data or data that is compressed using gzip. – **ORC**: DLI can read non-compressed data or data that is compressed using Snappy. – **JSON**: DLI can read non-compressed data or data that is compressed using gzip. – **Avro**: DLI can read uncompressed Avro data.	CSV
Storage Path	Enter or select an OBS path. The path can be a folder or a path. **NOTE** If you need to import data stored in OBS to the OBS table, set this parameter to the path of a folder. If the table creation path is a file, data fails to be imported.	obs://obs1/sampledata.csv
Table Header: No/Yes	This parameter is valid only when **Data Format** is set to **CSV**. Whether the data source to be imported contains the table header. Click **Advanced Settings** and select the check box next to **Table Header: No**. If the check box is selected, the table header is displayed. If the check box is deselected, no table header is displayed.	-
User-defined Delimiter	This parameter is valid only when **Data Format** is set to **CSV** and you select **User-defined Delimiter**. The following delimiters are supported: – Comma (,) – Vertical bar (|) – Tab character (\t) – Others: Enter a user-defined delimiter.	Comma (,)

Parameter	Description	Example
User-defined Quotation Character	This parameter is valid only when **Data Format** is set to **CSV** and you select **User-defined Quotation Character**. The following quotation characters are supported: – Single quotation mark (') – Double quotation marks (") – Others: Enter a user-defined quotation character.	Single quotation mark (')
User-defined Escape Character	This parameter is valid only when **Data Format** is set to **CSV** and you select **User-defined Escape Character**. The following escape characters are supported: – Backslash (\) – Others: Enter a user-defined escape character.	Backslash (\)
Date Format	This parameter is valid only when **Data Format** is set to **CSV** or **JSON**. This parameter specifies the format of the date in the table and is valid only **Advanced Settings** is selected. The default value is **yyyy-MM-dd**. For definition of characters involved in the date pattern, see Table 3 in .	2000-01-01
Timestamp Format	This parameter is valid only when **Data Format** is set to **CSV** or **JSON**. This parameter specifies the format of the timestamp in the table and is valid only **Advanced Settings** is selected. The default value is **yyyy-MM-dd HH:mm:ss**. For definition of characters involved in the time pattern, see Table 3 in .	2000-01-01 09:00:00

Step 3 Click **OK**.

After a table is created, you can view and select the table for use on the **Data Management** page or **SQL Editor** page.

Step 4 (Optional) After a DLI table is created, you can decide whether to directly import data to the table.

----End

7.1.5 Deleting a Database or a Table

You can delete unnecessary databases and tables based on actual conditions.

Precautions

- You are not allowed to delete databases or tables that are being used for running jobs.
- The administrator, database owner, and users with the database deletion permission can delete the database. The administrator, database owner, and users with the table deletion permission can delete the table.

  NOTE

  If a database or table is deleted, it cannot be recovered. Exercise caution when performing this operation.

Deleting a Table

You can delete a table on either the **Data Management** page or the **SQL Editor** page.

- Delete the table on the **Data Management** page.
  a. On the left of the management console, choose **Data Management** > **Databases and Tables**.
  b. Locate the row where the database whose tables you want to delete, click the database name to switch to the **Table Management** page.
  c. Locate the row where the target table locates and click **More** > **Delete** in the **Operation** column.
  d. In the displayed dialog box, click **Yes**.
- Delete a table on the **SQL Editor** page.
  a. On the top menu bar of the DLI management console, click **SQL Editor**.
  b. In the navigation tree on the left, click **Databases**. Click the name of a database where the table you want belongs. The tables of the selected database are displayed.
  c. Click ☰ on the right of the table and choose **Delete** from the shortcut menu.
  d. In the dialog box that is displayed, click **OK**.

Deleting a Database

1. On the left of the management console, choose **Data Management** > **Databases and Tables**.
2. Locate the row where the target database locates and click **More** > **Drop Database** in the **Operation** column.

   NOTE

   You cannot delete databases that contain tables. To delete a database containing tables, delete the tables first.
3. In the displayed dialog box, click **Yes**.

7.1.6 Modifying the Owners of Databases and Tables

During actual use, developers create databases and tables and submit them to test personnel for testing. After the test is complete, the databases and tables are transferred to O&M personnel for user experience. In this case, you can change the owner of the databases and tables to transfer data to other owners.

Modifying the Database Owner

You can change the owner of a database on either the **Data Management** page or the **SQL Editor** page.

- On the **Data Management** page, change the database owner.
  a. On the left of the management console, choose **Data Management** > **Databases and Tables**.
  b. On the **Databases and Tables** page, locate the database you want and click **More** > **Modify Database** in the **Operation** column.
  c. In the displayed dialog box, enter a new owner name (an existing username) and click **OK**.
- Change the database owner on the **SQL Editor** page.
  a. On the left of the management console, click **SQL Editor**.
  b. In the navigation tree on the left, click **Databases**, click on the right of the database you want to modify, and choose **Modify Database** from the shortcut menu.
  c. In the displayed dialog box, enter a new owner name (an existing username) and click **OK**.

Modifying the Table Owner

1. On the left of the management console, choose **Data Management** > **Databases and Tables**.
2. Click the name of the database corresponding to the table to be modified. The **Manage Tables** page of the database is displayed.
3. In the **Operation** column of the target table, choose **More** > **Modify Owner**.
4. In the displayed dialog box, enter a new owner name (an existing username) and click **OK**.

7.1.7 Importing Data to the Table

You can import data from OBS to a table created in DLI.

Precautions

- Only one path can be specified during data import. The path cannot contain commas (,).
- To import data in CSV format to a partitioned table, place the column to be partitioned in the last column of the data source.
- You are advised not to concurrently import data in to a table. If you concurrently import data into a table, there is a possibility that conflicts occur, leading to failed data import.

- The imported file can be in CSV, Parquet, ORC, JSON, and Avro format. The encoding format must be UTF-8.

Prerequisites

The data to be imported has been stored on OBS.

Procedure

Step 1 You can import data on either the **Data Management** page or the **SQL Editor** page.

- To import data on the **Data Management** page:
  a. On the left of the management console, choose **Data Management** > **Databases and Tables**.
  b. Click the name of the database corresponding to the table where data is to be imported to switch to the table management page.
  c. Locate the row where the target table resides and choose **More** > **Import** in the **Operation** column. The **Import** dialog box is displayed.
- To import data on the **SQL Editor** page:
  a. On the left of the management console, click **SQL Editor**.
  b. In the navigation tree on the left of **SQL Editor**, click **Databases** to see all databases. Click the database where the target table belongs. The table list is displayed.
  c. Click  on the right of the table and choose **Import** from the shortcut menu. The **Import** page is displayed.

Step 2 In the **Import** dialog box, set the parameters based on **Table 7-10**.

Table 7-10 Description

Parameter	Description	Example
Databases	Database where the current table is located.	-
Table Name	Name of the current table.	-
Queues	Queue where the imported data will be used	-
File Format	Format of the data source file to be imported. The CSV, Parquet, ORC, JSON, and Avro formats are supported. Encoding format. Only UTF-8 is supported.	CSV

Parameter	Description	Example
Path	You can directly enter a path or click  and select an OBS path. If no bucket is available, you can directly switch to the OBS management console and create an OBS bucket. • **When creating an OBS table, you must specify a folder as the directory. If a file is specified, data import may be failed.** • **If a folder and a file have the same name in the OBS directory, the file path is preferred as the path of the data to be imported.** **NOTE** The path can be a file or folder.	obs://DLI/sampledata.csv
Table Header: No/Yes	This parameter is valid only when **File Format** is set to **CSV**. Whether the data source to be imported contains the table header. Click **Advanced Settings** and select the check box next to **Table Header: No**. If the check box is selected, the table header is displayed. If the check box is deselected, no table header is displayed.	-
User-defined Delimiter	This parameter is valid only when **File Format** is set to **CSV** and you select **User-defined Delimiter**. The following delimiters are supported: • Comma (,) • Vertical bar (|) • Tab character (\t) • Others: Enter a user-defined delimiter.	Default value: comma (,)
User-defined Quotation Character	This parameter is valid only when **File Format** is set to **CSV** and **User-defined Quotation Character** is selected. The following quotation characters are supported: • Single quotation mark (') • Double quotation marks (") • Others: Enter a user-defined quotation character.	Default value: double quotation marks (")
User-defined Escape Character	This parameter is valid only when **File Format** is set to **CSV** and you select **User-defined Escape Character**. The following escape characters are supported: • Backslash (\) • Others: Enter a user-defined escape character.	Default value: backslash (\)

Parameter	Description	Example
Date Format	This parameter is valid only when **File Format** is set to **CSV** or **JSON**. This parameter specifies the format of the date in the table and is valid only **Advanced Settings** is selected. The default value is **yyyy-MM-dd**. For definition of characters involved in the date pattern, see Table 3 in .	2000-01-01
Timestamp Format	This parameter is valid only when **File Format** is set to **CSV** or **JSON**. This parameter specifies the format of the timestamp in the table and is valid only **Advanced Settings** is selected. The default value is **yyyy-MM-dd HH:mm:ss**. For definition of characters involved in the time pattern, see Table 3 in .	2000-01-01 09:00:00
Error Records Path	This parameter is valid only when **File Format** is set to **CSV** or **JSON**. The parameter specifies the error data is stored in the corresponding OBS path and is valid only **Advanced Settings** is selected.	obs://DLI/

Step 3 Click **OK**.

Step 4 You can view the imported data in either of the following ways:

NOTE

Currently, only the first 10 records are displayed.

- Choose **Data Management** > **Databases and Tables** in the navigation pane of the console. Locate the row that contains the database where the target table belongs and click **More** > **View Properties** in the **Operation** column. In the displayed dialog box, click the **Preview** tab to view the imported data.
- In the **Databases** tab of the **SQL Editor**, click the database name to go to the table list. Click on the right of a table name and choose **View Properties** from the shortcut menu. In the displayed dialog box, click **Preview** to view the imported data.

Step 5 (Optional) View the status and execution result of the importing job on the **Job Management** > **SQL Jobs** page.

----End

7.1.8 Exporting Data from DLI to OBS

You can export data from a DLI table to OBS. During the export, a folder is created in OBS or the content in the existing folder is overwritten.

Precautions

- The exported file can be in JSON format, and the text format can only be UTF-8.
- Only the data in the DLI table (the table type is **Managed**) can be exported to the OBS bucket, and the export path must be specified to the folder level.
- Data can be exported across accounts. That is, after account B authorizes account A, account A can export data to the OBS path of account B if account A has the permission to read the metadata and permission information about the OBS bucket of account B and read and write the path.

Procedure

Step 1 You can export data on either the **Data Management** page or the **SQL Editor** page.

- To export data on the **Data Management** page:
  a. On the left of the management console, choose **Data Management** > **Databases and Tables**.
  b. Click the name of the database corresponding to the table where data is to be exported to switch to the **Manage Tables** page.
  c. Select the corresponding table (DLI table) and choose **More** > **Export** in the **Operation** column. The **Export Data** page is displayed.
- To export data on the **SQL Editor** page:
  a. On the left of the management console, click **SQL Editor**.
  b. In the navigation tree on the left, click **Databases** to see all databases. Click the database name corresponding to the table to which data is to be exported. The tables are displayed.
  c. Click  on the right of the managed table (DLI table) whose data is to be exported, and choose **Export** from the shortcut menu.

Step 2 In the displayed **Export Data** dialog box, specify parameters by referring to **Table 7-11**.

Table 7-11 Parameter description

Parameter	Description
Databases	Database where the current table is located.
Table Name	Name of the current table.
Data Format	Format of the file storing data to be exported. Formats other than JSON will be supported in later versions.
Queue	Select a queue.

Parameter	Description
Compression Format	Compression format of the data to be exported. The following compression formats are supported: • none • bzip2 • deflate • gzip
Storage Path	• Enter or select an OBS path. • The export path must be a folder that does not exist in the OBS bucket. Specifically, you need to create a folder in the target OBS directory. • The folder name cannot contain the special characters of **\/:*?"<>|**, and cannot start or end with a dot (.).
Export Mode	Storage mode of the data to be exported. • **New OBS directory**: If the specified export directory exists, an error is reported and the export operation cannot be performed. • **Existing OBS directory (Overwritten)**: If you create a file in the specified directory, the existing file will be overwritten.
Table Header: No/Yes	Whether the data to be exported contains the table header.

Step 3 Click **OK**.

Step 4 (Optional) You can view the job status (indicated by **Status**), statements (indicated by **Statement**), and other information about exporting jobs on the **SQL Jobs** page.

1. Select **EXPORT** from the **Job Type** drop-down list box and specify the time range for exporting data. The jobs meeting the requirements are displayed in the job list.
2. Click ⌄ to view details about an exporting job.

----End

7.1.9 Viewing Metadata

Metadata Description

- Metadata is used to define data types. It describes information about the data, including the source, size, format, and other data features. In database fields, metadata interprets data content in the data warehouse.
- When you create a table, metadata is defined, consisting of the column name, type, and description.
- The **Metadata** page displays information about the target table, including **Column Name**, **Column Type**, **Data Type**, and **Description**.

Procedure

You can view metadata on either the **Data Management** page or the **SQL Editor** page.

- To view metadata on the **Data Management** page:
  a. On the left of the management console, choose **Data Management** > **Databases and Tables**.
  b. On the displayed **Data Management** page, click the name of the database where the target table whose data you want to export resides to switch to the **Manage Tables** page.
  c. Click **More** in the **Operation** column of the target table and select **View Properties**. In the **Metadata** tab, view the metadata of the table.
- To view metadata on the **SQL Editor** page:
  a. On the left of the management console, click **SQL Editor**.
  b. In the navigation pane of the displayed **SQL Editor** page, click **Databases**.
  c. Click the corresponding database name to view the tables in the database.
  d. Click on the right of the table and choose **View Properties** from the shortcut menu. On the **Metadata** tab page, view the metadata of the table.

7.1.10 Previewing Data

The **Preview** page displays the first 10 records in the table.

Procedure

You can preview data on either the **Data Management** page or the **SQL Editor** page.

- To preview data on the **Data Management** page:
  a. On the left of the management console, choose **Data Management** > **Databases and Tables**.
  b. On the displayed **Data Management** page, click the name of the database where the target table whose data you want to export resides to switch to the **Manage Tables** page.
  c. Click **More** in the **Operation** column of the target table and select **View Properties**.
  d. Click the **Preview** tab to preview the table data.
- To preview data on the **SQL Editor** page:
  a. On the left of the management console, click **SQL Editor**.
  b. In the navigation pane of the displayed **SQL Editor** page, click **Databases**.
  c. Click the corresponding database name to view the tables in the database.

d. Click on the right of the corresponding table, choose **View Properties** from the list menu, and click the **Preview** tab to preview the data of the table.

7.1.11 Managing Tags

Tag Management

A tag is a key-value pair that you can customize to identify cloud resources. It helps you to classify and search for cloud resources. A tag consists of a tag key and a tag value. If you use tags in other cloud services, you are advised to create the same tag (key-value pairs) for cloud resources used by the same business to keep consistency.

DLI supports the following two types of tags:

- Resource tags: non-global tags created on DLI
- Predefined tags: global tags created on Tag Management Service (TMS).

This section describes how to add, modify, and delete tags for databases and tables.

Database Tags

Step 1 In the navigation pane on the left, choose **Data Management** > **Databases and Tables**.

Step 2 Locate the row that contains the target database, and click **More** > **Tags** in the **Operation** column.

Step 3 The tag management page is displayed, and the tags (if there are) are displayed.

Step 4 On the displayed page, click **Add/Edit Tag**. The **Add/Edit Tag** dialog box is displayed.

Enter a tag key and a tag value in the text boxes and click **Add**.

Table 7-12 Tag parameters

Parameter	Description
Tag key	You can specify the tag key in either of the following ways: ● Click the text box for tag key and select a predefined tag key from the drop-down list. To add a predefined tag, you need to create one on TMS and then select it from the **Tag key** drop-down list. You can click **View predefined tags** to go to the **Predefined Tags** page of the TMS console. Then, click **Create Tag** in the upper corner of the page to create a predefined tag. ● Enter a tag key in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with _**sys**_.
Tag value	You can specify the tag value in either of the following ways: ● Click the tag value text box and select a predefined tag value from the drop-down list. ● Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 225 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

NOTE

- A maximum of 20 tags can be added.
- Only one tag value can be added to a tag key.
- The key name in each resource must be unique.

Step 5 Click **OK**. The database tag is added.

To delete a tag, click **Delete** in the **Operation** column of the target tag.

----End

Table Tags

Step 1 In the navigation pane on the left, choose **Data Management** > **Databases and Tables**.

Step 2 Click a database name to view the tables in the database.

Step 3 Locate the row that contains the target table and click **More** > **Tag** in the **Operation** column.

Step 4 The tag management page is displayed, and the tags (if there are) are displayed.

Step 5 On the displayed page, click **Add/Edit Tag**. The **Add/Edit Tag** dialog box is displayed.

Enter a tag key and a tag value in the text boxes and click **Add**

Table 7-13 Tag parameters

Parameter	Description
Tag key	You can specify the tag key in either of the following ways: • Click the text box for tag key and select a predefined tag key from the drop-down list. To add a predefined tag, you need to create one on TMS and then select it from the **Tag key** drop-down list. You can click **View predefined tags** to go to the **Predefined Tags** page of the TMS console. Then, click **Create Tag** in the upper corner of the page to create a predefined tag. • Enter a tag key in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with _**sys**_.
Tag value	You can specify the tag value in either of the following ways: • Click the tag value text box and select a predefined tag value from the drop-down list. • Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 225 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

NOTE

- A maximum of 20 tags can be added.
- Only one tag value can be added to a tag key.
- The key name in each resource must be unique.

Step 6 Click **OK**. The table tag is added.

To delete a tag, click **Delete** in the **Operation** column of the target tag.

----End

7.2 Package Management

7.2.1 Overview

Package management provides the following functions:

- **Managing Package Permissions**
- **Creating a Package**

- **Deleting a Package**

  NOTE

  You can delete program packages in batches.

- **Modifying the Owner**

Constraints

- A package can be deleted, but a package group cannot be deleted.
- The following types of packages can be uploaded:
  - **JAR**: JAR file
  - **PyFile**: User Python file
  - **File**: User file
  - **ModelFile**: User AI model file

Package Management Page

Table 7-14 Package management parameters

Parameter	Description
Group Name	Name of the group to which the package belongs. If the package is not grouped, -- is displayed.
Package Name	Name of a package.
Owner	Name of the user who uploads the package.
Type	Type of a package. The following package types are supported: • JAR: JAR file • PyFile: User Python file • File: User file
Status	Status of the package to be created. • Uploading: The file is being uploaded. • Finished: The resource package has been uploaded. • Failed: The resource package upload failed.
Created	Time when a package is created.
Updated	Time when the package is updated.
Operation	Manage Permissions: Manage user permissions for a package. Delete: Delete the package. **More**: • **Modify Owner**: Modify the owner of the package.

7.2.2 Managing Permissions on Packages and Package Groups

Scenario

- You can isolate package groups or packages allocated to different users by setting permissions to ensure data query performance.
- The administrator and the owner of a package group or package have all permissions. You do not need to set permissions and the permissions cannot be modified by other users.
- When you set permissions on a package group or a package to a new user, the user group the user belong to must have the Tenant Guest permission.

On the **Package Management** page, click **Manage Permissions** in the **Operation** column of the target package. On the displayed **User Permission Info** page, you can grant permissions for the package group or package, set and revoke user permissions.

NOTE

- If you select a group when creating a package, you can manage permissions of the corresponding program package group.
- If you select **No grouping** when creating a package, you can manage permissions of the corresponding package.

Granting Permissions on Package Groups/Packages

Click **Grant Permission** in the upper right corner of the page.

- Granting permissions on package groups

Table 7-15 Permission parameters

Parameter	Description
Username	Name of the authorized IAM user. **NOTE** The username is the name of an existing IAM user.

Parameter	Description
Select the permissions to be granted to the user	• **Use Group**: This permission allows you to use the package of this group. • **Update Group**: This permission allows you to update the packages in the group, including creating a package in the group. • **Query Group**: This permission allows you to query the details of a package in a group. • **Delete Group**: This permission allows you to delete the package of the group. • **Grant Permission**: This permission allows you to grant group permissions to other users. • **Revoke Permission**: This permission allows you to revoke the group permissions that other users have but cannot revoke the group owner's permissions. • **View Other User's Permissions**: This permission allows you to view the group permissions of other users.

- Granting permissions on packages

Table 7-16 Permission parameters

Parameter	Description
Username	Name of the authorized IAM user. **NOTE** The username is the name of an existing IAM user.
Select the permissions to be granted to the user	• **Use Package**: This permission allows you to use the package. • **Update Package**: This permission allows you to update the package. • **Query Package**: This permission allows you to query the package. • **Delete Package**: This permission allows you to delete the package. • **Grant Permission**: This permission allows you to grant package permissions to other users. • **Revoke Permission**: This permission allows you to revoke the package permissions that other users have but cannot revoke the package owner's permissions. • **View Other User's Permissions**: This permission allows you to view the package permissions of other users.

Setting Permissions on Package Groups and Packages

Click **Set Permission** in the **Operation** column of the sub-user to modify the permission of the user. **Table 7-15** and **Table 7-16** list the detailed permission descriptions.

If the **Set Permission** button is gray, you do not have the permission to modify the package group or package. You can apply to the administrator, group owner, or other users who have the permissions on granting and revoking permissions of package groups or packages.

Revoking Permissions on Package Groups and Packages

Click **Revoke Permission** in the **Operation** column of a sub-user to revoke the user's permissions. After the operation, the sub-user does not have any permission on the package group or package.

Permissions Description

- Package group permissions

  Querying permissions. A group owner can view the created package group and all packages in the group, and can also view package groups on which they have all permissions.

  A package group is a unit. If you select a group when creating a package, you can grant only the permissions of the package group to other users.

- Package permissions

  Querying permissions. A package owner can view the created packages, and can also view packages on which they have all permissions.

7.2.3 Creating a Package

DLI allows you to submit program packages in batches to the general-use queue for running.

NOTE

If you need to update a package, you can use the same package or file to upload it to the same location (in the same group) on DLI to overwrite the original package or file.

Prerequisites

All software packages must be uploaded to OBS for storage in advance.

Procedure

1. On the left of the management console, choose **Data Management** > **Package Management**.
2. On the **Package Management** page, click **Create** in the upper right corner to create a package.
3. In the displayed **Create Package** dialog box, set related parameters by referring to **Table 7-17**.

Table 7-17 Parameter description

Parameter	Description
Package Type	The following package types are supported: • JAR: JAR file • PyFile: User Python file • File: User file
Package File Path	Select the OBS path of the corresponding packages. **NOTE** • The packages must be uploaded to OBS for storage in advance. • Only files can be selected.
Group Policy	You can select **Use existing group**, **Use new group**, or **No grouping**.
Group Name	• **Use existing group**: Select an existing group. • **Use new group**: Enter a custom group name. • **No grouping**: No need to select or enter a group name. **NOTE** • If you select a group, the permission management refers to the permissions of the corresponding package group. • If no group is selected, the permission management refers to the permissions of the corresponding package. For details about how to manage permissions on package groups and packages, see **Managing Permissions on Packages and Package Groups**.
Tag	Tags used to identify cloud resources. A tag includes the tag key and tag value. If you want to use the same tag to identify multiple cloud resources, that is, to select the same tag from the drop-down list box for all services, you are advised to create predefined tags on the Tag Management Service (TMS). **NOTE** • A maximum of 20 tags can be added. • Only one tag value can be added to a tag key. • The key name in each resource must be unique. • Tag key: Enter a tag key name in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with _**sys**_. • Tag value: Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 225 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

4. Click **OK**.

After a package is created, you can view and select the package for use on the **Package Management** page.

7.2.4 Deleting a Package

You can delete a package based on actual conditions.

Procedure

1. On the left of the management console, choose **Data Management** > **Package Management**.
2. Click **Delete** in the **Operation** column of the package to be deleted.
3. In the dialog box that is displayed, click **Yes**.

7.2.5 Modifying the Owner

To change the owner of a package, click **More** > **Modify Owner** in the **Operation** column of a package on the **Package Management** page.

- If the package has been grouped, you can modify the owner of the **Group** or **Resource** of it.
- If the package has not been grouped, change its owner directly.

Table 7-18 Description

Parameter	Description
Group Name	• If you select a group when creating a package, the name of the group is displayed. • If no group is selected when creating a package, this parameter is not displayed.
Name	Name of a package.
Select Type	• If you select a group when creating a package, you can change the owner of the group or package. • If no group is selected when creating a package, this parameter is not displayed.
Username	Name of the package owner. **NOTE** The username is the name of an existing IAM user.

7.2.6 Built-in Dependencies

DLI built-in dependencies are provided by the platform by default. In case of conflicts, you do not need to upload them when packaging JAR packages of Spark or Flink Jar jobs.

Spark 2.3.2 Dependencies

- accessors-smart-1.2.jar
- activation-1.1.1.jar
- aircompressor-0.8.jar
- alluxio-2.3.1-luxor-SNAPSHOT-client.jar
- antlr-2.7.7.jar
- antlr4-runtime-4.8-1.jar
- antlr-runtime-3.4.jar
- aopalliance-1.0.jar
- aopalliance-repackaged-2.4.0-b34.jar
- apache-log4j-extras-1.2.17.jar
- arpack_combined_all-0.1.jar
- arrow-format-0.8.0.jar
- arrow-memory-0.8.0.jar
- arrow-vector-0.8.0.jar
- asm-5.0.4.jar
- audience-annotations-0.5.0.jar
- automaton-1.11-8.jar
- avro-1.7.7.jar
- avro-ipc-1.7.7.jar
- avro-ipc-1.7.7-tests.jar
- avro-mapred-1.7.7-hadoop2.jar
- java-sdk-bundle-1.11.271.jar
- base64-2.3.8.jar
- bcpkix-jdk15on-1.66.jar
- bcprov-jdk15on-1.66.jar
- bonecp-0.8.0.RELEASE.jar
- breeze_2.11-0.13.2.jar
- breeze-macros_2.11-0.13.2.jar
- calcite-avatica-1.2.0-incubating.jar
- calcite-core-1.2.0-incubating.jar
- calcite-linq4j-1.2.0-incubating.jar
- checker-qual-2.11.1.jar
- chill_2.11-0.8.4.jar
- chill-java-0.8.4.jar
- commons-beanutils-1.9.4.jar
- commons-cli-1.2.jar
- commons-codec-2.0-20130428.202122-59.jar
- commons-collections-3.2.2.jar
- commons-collections4-4.2.jar

- commons-compiler-3.0.8.jar
- commons-compress-1.4.1.jar
- commons-configuration2-2.1.1.jar
- commons-crypto-1.0.0-20191105.jar
- commons-daemon-1.0.13.jar
- commons-dbcp-1.4.jar
- commons-dbcp2-2.7.0.jar
- commons-httpclient-3.1.jar
- commons-io-2.5.jar
- commons-lang-2.6.jar
- commons-lang3-3.5.jar
- commons-logging-1.2.jar
- commons-math3-3.4.1.jar
- commons-net-2.2.jar
- commons-pool-1.5.4.jar
- commons-pool2-2.8.0.jar
- commons-text-1.3.jar
- compress-lzf-1.0.3.jar
- core-1.1.2.jar
- curator-client-4.2.0.jar
- curator-framework-4.2.0.jar
- curator-recipes-2.7.1.jar
- datanucleus-api-jdo-3.2.6.jar
- datanucleus-core-3.2.10.jar
- datanucleus-rdbms-3.2.9.jar
- derby-10.12.1.1.jar
- dnsjava-2.1.7.jar
- ehcache-3.3.1.jar
- eigenbase-properties-1.1.5.jar
- error_prone_annotations-2.3.4.jar
- failureaccess-1.0.1.jar
- fastutil-8.2.3.jar
- ffmpeg-4.3.1-1.5.4.jar
- ffmpeg-4.3.1-1.5.4-linux-x86_64.jar
- flatbuffers-1.2.0-3f79e055.jar
- generex-1.0.2.jar
- geronimo-jcache_1.0_spec-1.0-alpha-1.jar
- gson-2.2.4.jar
- guava-29.0-jre.jar
- guice-4.0.jar

- guice-servlet-4.0.jar
- hadoop-annotations-3.1.1-ei-302002.jar
- hadoop-auth-3.1.1-ei-302002.jar
- hadoop-3.1.1-ei-302002.jar
- hadoop-client-3.1.1-ei-302002.jar
- hadoop-common-3.1.1-ei-302002.jar
- hadoop-hdfs-3.1.1-ei-302002.jar
- hadoop-hdfs-client-3.1.1-ei-302002.jar
- hadoop-mapreduce-client-common-3.1.1-ei-302002.jar
- hadoop-mapreduce-client-core-3.1.1-ei-302002.jar
- hadoop-mapreduce-client-jobclient-3.1.1-ei-302002.jar
- hadoop-minikdc-3.1.1-ei-302002.jar
- hadoop-yarn-api-3.1.1-ei-302002.jar
- hadoop-yarn-client-3.1.1-ei-302002.jar
- hadoop-yarn-common-3.1.1-ei-302002.jar
- hadoop-yarn-registry-3.1.1-ei-302002.jar
- hadoop-yarn-server-common-3.1.1-ei-302002.jar
- hadoop-yarn-server-web-proxy-3.1.1-ei-302002.jar
- hamcrest-core-1.3.jar
- HikariCP-java7-2.4.12.jar
- hive-common-1.2.1-2.1.0.dli-20201111.064115-91.jar
- hive-exec-1.2.1-2.1.0.dli-20201111.064444-91.jar
- hive-metastore-1.2.1-2.1.0.dli-20201111.064230-91.jar
- hk2-api-2.4.0-b34.jar
- hk2-locator-2.4.0-b34.jar
- hk2-utils-2.4.0-b34.jar
- hppc-0.7.2.jar
- htrace-core4-4.2.0-incubating-1.0.0.jar
- httpclient-4.5.4.jar
- httpcore-4.4.7.jar
- ivy-2.4.0.jar
- j2objc-annotations-1.3.jar
- jackson-annotations-2.10.0.jar
- jackson-core-2.10.0.jar
- jackson-core-asl-1.9.13-atlassian-4.jar
- jackson-databind-2.10.0.jar
- jackson-dataformat-yaml-2.10.0.jar
- jackson-datatype-jsr310-2.10.3.jar
- jackson-jaxrs-base-2.10.3.jar
- jackson-jaxrs-json-provider-2.10.3.jar

- jackson-mapper-asl-1.9.13-atlassian-4.jar
- jackson-module-jaxb-annotations-2.10.3.jar
- jackson-module-paranamer-2.10.0.jar
- jackson-module-scala_2.11-2.10.0.jar
- jakarta.activation-api-1.2.1.jar
- jakarta.xml.bind-api-2.3.2.jar
- janino-3.0.8.jar
- javacpp-1.5.4.jar
- javacpp-1.5.4-linux-x86_64.jar
- javacv-1.5.4.jar
- JavaEWAH-1.1.7.jar
- javassist-3.18.1-GA.jar
- javax.annotation-api-1.2.jar
- javax.inject-1.jar
- javax.inject-2.4.0-b34.jar
- javax.servlet-api-3.1.0.jar
- javax.ws.rs-api-2.0.1.jar
- java-xmlbuilder-1.1.jar
- javolution-5.3.1.jar
- jaxb-api-2.2.11.jar
- jcip-annotations-1.0-1.jar
- jcl-over-slf4j-1.7.26.jar
- jdo-api-3.0.1.jar
- jersey-client-2.23.1.jar
- jersey-common-2.23.1.jar
- jersey-container-servlet-2.23.1.jar
- jersey-container-servlet-core-2.23.1.jar
- jersey-guava-2.23.1.jar
- jersey-media-jaxb-2.23.1.jar
- jersey-server-2.23.1.jar
- jets3t-0.9.4.jar
- jetty-http-9.4.31.v20200723.jar
- jetty-io-9.4.31.v20200723.jar
- jetty-security-9.4.31.v20200723.jar
- jetty-server-9.4.31.v20200723.jar
- jetty-servlet-9.4.31.v20200723.jar
- jetty-util-9.4.31.v20200723.jar
- jetty-util-ajax-9.4.31.v20200723.jar
- jetty-webapp-9.4.31.v20200723.jar
- jetty-xml-9.4.31.v20200723.jar

- joda-time-2.9.3.jar
- jodd-core-4.2.0.jar
- json-20200518.jar
- json4s-ast_2.11-3.2.11.jar
- json4s-core_2.11-3.2.11.jar
- json4s-jackson_2.11-3.2.11.jar
- json-sanitizer-1.2.1.jar
- json-smart-2.3.jar
- jsp-api-2.1.jar
- jsr305-3.0.2.jar
- jta-1.1.jar
- jtransforms-2.4.0.jar
- jul-to-slf4j-1.7.26.jar
- junit-4.11.jar
- kerb-admin-1.0.1.jar
- kerb-client-1.0.1.jar
- kerb-common-1.0.1.jar
- kerb-core-1.0.1.jar
- kerb-crypto-1.0.1.jar
- kerb-identity-1.0.1.jar
- kerb-server-1.0.1.jar
- kerb-simplekdc-1.0.1.jar
- kerb-util-1.0.1.jar
- kerby-asn1-1.0.1.jar
- kerby-config-1.0.1.jar
- kerby-pkix-1.0.1.jar
- kerby-util-1.0.1.jar
- kerby-xdr-1.0.1.jar
- kryo-shaded-3.0.3.jar
- kubernetes-client-4.9.2-20200804.jar
- kubernetes-model-4.9.2-20200804.jar
- kubernetes-model-common-4.9.2-20200804.jar
- leveldbjni-all-1.8-20191105.jar
- libfb303-0.9.3.jar
- libthrift-0.12.0.jar
- listenablefuture-9999.0-empty-to-avoid-conflict-with-guava.jar
- log4j-1.2.17-cloudera1.jar
- log4j-rolling-appender-20131024-2017.jar
- logging-interceptor-3.14.4.jar
- luxor-encrypt-2.1.0-20201106.065437-53.jar

- luxor-fs3-2.1.0-20201106.065612-53.jar
- luxor-obs-fs3-2.1.0-20201106.065616-53.jar
- luxor-rpc_2.11-2.1.0-20201106.065541-53.jar
- luxor-rpc-protobuf2-2.1.0-20201106.065551-53.jar
- lz4-java-1.7.1.jar
- machinist_2.11-0.6.1.jar
- macro-compat_2.11-1.1.1.jar
- metrics-core-3.1.5.jar
- metrics-graphite-3.1.5.jar
- metrics-jmx-4.1.12.1.jar
- metrics-json-3.1.5.jar
- metrics-jvm-3.1.5.jar
- minlog-1.3.0.jar
- mssql-jdbc-6.2.1.jre7.jar
- netty-3.10.6.Final.jar
- netty-all-4.1.51.Final.jar
- nimbus-jose-jwt-8.19.jar
- objenesis-2.1.jar
- okhttp-3.14.4.jar
- okio-1.17.2.jar
- opencsv-2.3.jar
- opencsv-4.6.jar
- opencv-4.3.0-2.jar
- orc-core-1.4.4-nohive.jar
- orc-mapreduce-1.4.4-nohive.jar
- oro-2.0.8.jar
- osgi-resource-locator-1.0.1.jar
- paranamer-2.8.jar
- parquet-column-1.8.3.jar
- parquet-common-1.8.3.jar
- parquet-encoding-1.8.3.jar
- parquet-format-2.3.1.jar
- parquet-hadoop-1.8.3.jar
- parquet-hadoop-bundle-1.6.0.jar
- parquet-jackson-1.8.3.jar
- parquet-format-2.3.1.jar
- parquet-hadoop-1.8.3.jar
- parquet-hadoop-bundle-1.6.0.jar
- parquet-jackson-1.8.3.jar
- postgresql-42.2.14.jar

- protobuf-java-2.5.0.jar
- py4j-0.10.7.jar
- pyrolite-4.13.jar
- re2j-1.1.jar
- RoaringBitmap-0.5.11.jar
- scala-compiler-2.11.12.jar
- scala-library-2.11.12.jar
- scalap-2.11.0.jar
- scala-parser-combinators_2.11-1.1.0.jar
- scala-reflect-2.11.12.jar
- scala-xml_2.11-1.0.5.jar
- secComponentApi-1.0.5c.jar
- shapeless_2.11-2.3.2.jar
- slf4j-api-1.7.30.jar
- slf4j-log4j12-1.7.30.jar
- snakeyaml-1.24.jar
- snappy-java-1.1.7.5.jar
- spark-catalyst_2.11-2.3.2.0101-2.1.0.dli-20201111.073826-143.jar
- spark-core_2.11-2.3.2.0101-.0.dli-20201111.073836-134.jar
- spark-graphx_2.11-2.3.2.0101-2.1.0.dli-20201111.073847-129.jar
- spark-hive_2.11-2.3.2.0101-.0.dli-20201111.073854-132.jar
- spark-kubernetes_2.11-2.3.2.0101-2.1.0.dli-20201111.073916-85.jar
- spark-kvstore_2.11-2.3.2.0101-2.1.0.dli-20201111.073933-127.jar
- spark-launcher_2.11-2.3.2.0101-2.1.0.dli-20201111.073940-127.jar
- spark-mllib_2.11-2.3.2.0101-2.1.0.dli-20201111.073946-127.jar
- spark-mllib-local_2.11-2.3.2.0101-2.1.0.dli-20201111.073953-127.jar
- spark-network-common_2.11-2.3.2.0101-2.1.0.dli-20201111.073959-127.jar
- spark-network-shuffle_2.11-2.3.2.0101-2.1.0.dli-20201111.074007-127.jar
- spark-om_2.11-2.3.2.0101-.0.dli-20201111.074019-125.jar
- spark-repl_2.11-2.3.2.0101-2.1.0.dli-20201111.074028-125.jar
- spark-sketch_2.11-2.3.2.0101-2.1.0.dli-20201111.074035-125.jar
- spark-sql_2.11-2.3.2.0101-2.1.0.dli-20201111.074041-126.jar
- spark-streaming_2.11-2.3.2.0101-2.1.0.dli-20201111.074100-123.jar
- spark-tags_2.11-2.3.2.0101-2.1.0.dli-20201111.074136-123.jar
- spark-tags_2.11-2.3.2.0101-2.1.0.dli-20201111.074141-124-tests.jar
- spark-unsafe_2.11-2.3.2.0101-2.1.0.dli-20201111.074144-123.jar
- spark-uquery_2.11-2.3.2.0101-2.1.0.dli-20201111.074906-210.jar
- spark-yarn_2.11-2.3.2.0101-2.1.0.dli-20201111.074151-123.jar
- spire_2.11-0.13.0.jar
- spire-macros_2.11-0.13.0.jar

- ST4-4.3.1.jar
- stax2-api-3.1.4.jar
- stax-api-1.0-2.jar
- stream-2.7.0.jar
- stringtemplate-3.2.1.jar
- token-provider-1.0.1.jar
- univocity-parsers-2.5.9.jar
- validation-api-1.1.0.Final.jar
- woodstox-core-5.0.3.jar
- xbean-asm5-shaded-4.4.jar
- xercesImpl-2.12.0.jar
- xml-apis-1.4.01.jar
- xz-1.0.jar
- zjsonpatch-0.3.0.jar
- zookeeper-3.5.6-ei-302002.jar
- zookeeper-jute-3.5.6-ei-302002.jar
- zstd-jni-1.4.4-11.jar

Flink 1.7.2 Dependencies

- bcpkix-jdk15on-1.60.jar
- bcprov-jdk15on-1.60.jar
- commons-codec-1.9.jar
- commons-configuration-1.7.jar
- deeplearning4j-core-0.9.1.jar
- deeplearning4j-nlp-0.9.1.jar
- deeplearning4j-nn-0.9.1.jar
- ejml-cdense-0.33.jar
- ejml-core-0.33.jar
- ejml-ddense-0.33.jar
- ejml-dsparse-0.33.jar
- ejml-experimental-0.33.jar
- ejml-fdense-0.33.jar
- ejml-simple-0.33.jar
- ejml-zdense-0.33.jar
- elsa-3.0.0-M7.jar
- esdk-obs-java-3.1.3.jar
- flink-cep_2.11-1.7.0.jar
- flink-cep-scala_2.11-1.7.0.jar
- flink-dist_2.11-1.7.0.jar
- flink-gelly_2.11-1.7.0.jar

- flink-gelly-scala_2.11-1.7.0.jar
- flink-ml_2.11-1.7.0.jar
- flink-python_2.11-1.7.0.jar
- flink-queryable-state-runtime_2.11-1.7.0.jar
- flink-shaded-curator-1.7.0.jar
- flink-shaded-hadoop2-uber-1.7.0.jar
- flink-table_2.11-1.7.0.jar
- guava-26.0-jre.jar
- hadoop-3.1.1-41-20201014.085840-4.jar
- httpasyncclient-4.1.2.jar
- httpclient-4.5.12.jar
- httpcore-4.4.4.jar
- httpcore-nio-4.4.4.jar
- java-xmlbuilder-1.1.jar
- jna-4.1.0.jar
- libtensorflow-1.12.0.jar
- log4j-api-2.8.2.jar
- log4j-core-2.8.2.jar
- log4j-over-slf4j-1.7.21.jar
- logback-classic-1.2.3.jar
- logback-core-1.2.3.jar
- nd4j-api-0.9.1.jar
- nd4j-native-0.9.1.jar
- nd4j-native-api-0.9.1.jar
- nd4j-native-platform-0.9.1.jar
- okhttp-3.14.8.jar
- okio-1.14.0.jar
- slf4j-api-1.7.21.jar
- tensorflow-1.12.0.jar

Flink 1.10 Dependencies

Only queues created after December 2020 can use the Flink 1.10 dependencies.

- bcpkix-jdk15on-1.60.jar
- bcprov-jdk15on-1.60.jar
- commons-codec-1.9.jar
- commons-configuration-1.7.jar
- deeplearning4j-core-0.9.1.jar
- deeplearning4j-nlp-0.9.1.jar
- deeplearning4j-nn-0.9.1.jar
- ejml-cdense-0.33.jar

- ejml-core-0.33.jar
- ejml-ddense-0.33.jar
- ejml-dsparse-0.33.jar
- ejml-experimental-0.33.jar
- ejml-fdense-0.33.jar
- ejml-simple-0.33.jar
- ejml-zdense-0.33.jar
- elsa-3.0.0-M7.jar
- esdk-obs-java-3.20.6.1.jar
- flink-cep_2.11-1.10.0.jar
- flink-cep-scala_2.11-1.10.0.jar
- flink-dist_2.11-1.10.0.jar
- flink-python_2.11-1.10.0.jar
- flink-queryable-state-runtime_2.11-1.10.0.jar
- flink-sql-client_2.11-1.10.0.jar
- flink-state-processor-api_2.11-1.10.0.jar
- flink-table_2.11-1.10.0.jar
- flink-table-blink_2.11-1.10.0.jar
- guava-26.0-jre.jar
- hadoop-3.1.1-41.jar
- httpasyncclient-4.1.2.jar
- httpclient-4.5.3.jar
- httpcore-4.4.4.jar
- httpcore-nio-4.4.4.jar
- java-xmlbuilder-1.1.jar
- jna-4.1.0.jar
- libtensorflow-1.12.0.jar
- log4j-over-slf4j-1.7.26.jar
- logback-classic-1.2.3.jar
- logback-core-1.2.3.jar
- nd4j-api-0.9.1.jar
- nd4j-native-0.9.1.jar
- nd4j-native-api-0.9.1.jar
- nd4j-native-platform-0.9.1.jar
- okhttp-3.14.8.jar
- okio-1.14.0.jar
- secComponentApi-1.0.5.jar
- slf4j-api-1.7.26.jar
- tensorflow-1.12.0.jar

8 Job Templates

8.1 Managing SQL Templates

To facilitate SQL operation execution, DLI allows you to customize query templates or save the SQL statements in use as templates. After templates are saved, you do not need to compile SQL statements. Instead, you can directly perform the SQL operations using the templates.

SQL templates include sample templates and custom templates. The default sample template contains 22 standard TPC-H query statements, which can meet most TPC-H test requirements. For details, see **TPC-H Sample Data in the SQL Template**.

SQL template management provides the following functions:

- **Sample Templates**
- **Custom Templates**
- **Creating a Template**
- **Executing the Template**
- **Searching for a Template**
- **Modifying a Template**
- **Deleting a Template**

Table Settings

In the upper right corner of the **SQL Template** page, click **Set Property** to determine whether to display templates by group.

If you select **Display by Group**, the following display modes are available:

- Expand the first group
- Expand all
- Collapse All

Sample Templates

The current sample template contains 22 standard TPC-H query statements. You can view the template name, description, and statements. For details about TPC-H examples, see **TPC-H Sample Data in the SQL Template**.

Table 8-1 Template management parameters

Parameter	Description
Name	Indicates the template name. • A template name can contain only digits, letters, and underscores (_), but cannot start with an underscore (_) or contain only digits. It cannot be left empty. • The template name can contain a maximum of 50 characters.
Description	Description of the template you create.
Statement	SQL statement created as the template.
Operation	**Execute**: After you click this button, the system switches to the **SQL Editor** page, where you can modify or directly perform the statement as required. For details, see **Executing the Template**.

The existing sample templates apply to the following scenarios:

- Price summary report query
- Minimum cost supplier analysis
- Shipping priority analysis
- Analysis of order priority check
- Analysis of the number of local suppliers
- Analysis of forecasted income changes
- Freight volume analysis
- National market share analysis
- Profit estimation analysis by product type
- Analysis of returned parts
- Analysis of key inventory indicators
- Freight mode and command priority analysis
- Consumer allocation analysis
- Promotion effect analysis
- Analysis of the supplier with the largest contribution

- Analysis of the relationship between parts and suppliers
- Revenue analysis of small-lot orders
- Customer analysis for large orders
- Discounted revenue analysis
- Analysis of potential component improvements
- Analysis of suppliers who fail to deliver goods on time
- Global sales opportunity analysis

Custom Templates

The custom template list displays all templates you have created. You can view the template name, description, statements, and more.

Table 8-2 Template management parameters

Parameter	Description
Name	Indicates the template name. • A template name can contain only digits, letters, and underscores (_), but cannot start with an underscore (_) or contain only digits. It cannot be left empty. • The template name can contain a maximum of 50 characters.
Description	Description of the template you create.
Statement	SQL statement created as the template.
Operation	• **Execute**: After you click this button, the system switches to the **SQL Editor** page, where you can modify or directly perform the statement as required. For details, see **Executing the Template**. • **Modify**: Click **Modify**. In the displayed **Modify Template** dialog box, modify the template information as required. For details, see **Modifying a Template**.

Creating a Template

You can create a template on either the **Job Templates** or the **SQL Editor** page.

- To create a template on the **Job Templates** page:
  a. On the left of the management console, choose **Job Templates** > **SQL Templates**.
  b. On the **SQL Templates** page, click **Create Template** to create a template.

     Enter the template name, SQL statement, and description information. For details, see **Table 8-3**.

Table 8-3 Parameter description

Parameter	Description
Name	Indicates the template name. • A template name can contain only digits, letters, and underscores (_), but cannot start with an underscore (_) or contain only digits. It cannot be left empty. • The template name can contain a maximum of 50 characters.
Statement	SQL statement to be saved as a template.
Description	Description of the template you create.
Group	• Use existing • Use new • Do not use
Group Name	If you select **Use existing** or **Use new**, you need to enter the group name.

c. Click **OK**.

- To create a template on the **SQL Editor** page:

  a. On the left of the management console, click **SQL Editor**.

  b. In the SQL job editing area of the displayed **SQL Editor** page, click **More** in the upper right corner and choose **Save as Template**.

  Enter the template name, SQL statement, and description information. For details, see **Table 8-3**.

  c. Click **OK**.

Executing the Template

Perform the template operation as follows:

1. On the left of the management console, choose **Job Templates** > **SQL Templates**.
2. On the **SQL Templates** page, select a template and click **Execute** in the **Operation** column. The **SQL Editor** page is displayed, and the corresponding SQL statement is automatically entered in the SQL job editing window.
3. In the upper right corner of the SQL job editing window, Click **Execute** to run the SQL statement. After the execution is complete, you can view the execution result below the current SQL job editing window.

Searching for a Template

On the **SQL Templates** page, you can enter the template name keyword in the search box on the upper right corner to search for the desired template.

Modifying a Template

Only custom templates can be modified. To modify a template, perform the following steps:

Step 1 On the **SQL Templates** page, locate the target template and click **Modify** in the **Operation** column.

Step 2 In the displayed **Modify Template** dialog box, modify the template name, statement, and description as required.

Step 3 Click **OK**.

----End

Deleting a Template

On the **SQL Templates** page, select one or more templates to be deleted and click **Delete** to delete the selected templates.

8.2 Managing Flink Templates

Flink templates include sample templates and custom templates. You can modify an existing sample template to meet the actual job logic requirements and save time for editing SQL statements. You can also customize a job template based on your habits and methods so that you can directly invoke or modify the template in later jobs.

Flink template management provides the following functions:

- **Flink SQL Sample Template**
- **Custom Templates**
- **Creating a Template**
- **Creating a Job Based on a Template**
- **Modifying a Template**
- **Deleting a Template**

Flink SQL Sample Template

The template list displays existing sample templates for Flink SQL jobs. **Table 1** describes the parameters in the template list.

The scenarios of sample templates can be different, which are subject to the console.

Table 8-4 Parameters in the Flink SQL sample template list

Parameter	Description
Name	Name of a template, which has 1 to 64 characters and only contains letters, digits, hyphens (-), and underlines (_).
Description	Description of a template. It contains 0 to 512 characters.

Parameter	Description
Operation	**Create Job**: Create a job directly by using the template. After a job is created, the system switches to the **Edit** page under **Job Management**.

Custom Templates

The custom template list displays all Jar job templates. **Table 1** describes parameters in the custom template list.

Table 8-5 Parameters in the custom template list

Parameter	Description
Name	Name of a template, which has 1 to 64 characters and only contains letters, digits, hyphens (-), and underlines (_).
Description	Description of a template. It contains 0 to 512 characters.
Created	Time when a template is created.
Updated	Latest time when a template is modified.
Operation	• **Edit**: Modify a template that has been created. • **Create Job**: Create a job directly by using the template. After a job is created, the system switches to the **Edit** page under **Job Management**. • **More**: – **Delete**: Delete a created template.

Creating a Template

You can create a template using any of the following methods:

- Creating a template on the **Template Management** page
  a. In the left navigation pane of the DLI management console, choose **Job Templates** > **Flink Templates**.
  b. Click **Create Template** in the upper right corner of the page. The **Create Template** dialog box is displayed.
  c. Specify **Name** and **Description**.

Table 8-6 Template parameters

Parameter	Description
Name	Name of a template, which has 1 to 64 characters and only contains letters, digits, hyphens (-), and underlines (_). **NOTE** The template name must be unique.
Description	Description of a template. It contains 0 to 512 characters.
Tags	Tags used to identify cloud resources. A tag includes the tag key and tag value. If you want to use the same tag to identify multiple cloud resources, that is, to select the same tag from the drop-down list box for all services, you are advised to create predefined tags on the Tag Management Service (TMS). **NOTE** • A maximum of 20 tags can be added. • Only one tag value can be added to a tag key. • The key name in each resource must be unique. • Tag key: Enter a tag key name in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with _**sys**_. • Tag value: Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 225 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

d. Click **OK** to enter the editing page.

The **Table 8-7** describes the parameters on the template editing page.

Table 8-7 Template parameters

Parameter	Description
Name	You can modify the template name.
Description	You can modify the template description.
Saving Mode	• **Save Here**: Save the modification to the current template. • **Save as New**: Save the modification as a new template.

Parameter	Description
SQL statement editing area	In the area, you can enter detailed SQL statements to implement business logic. For details about how to compile SQL statements, see Data Lake Insight SQL Syntax Reference.
Save	Save the modifications.
Create Job	Use the current template to create a job.
Format	Format SQL statements. After SQL statements are formatted, you need to compile SQL statements again.
Theme Settings	Change the font size, word wrap, and page style (black or white background).

e. In the SQL statement editing area, enter SQL statements to implement service logic. For details about how to compile SQL statements, see .

f. After the SQL statement is edited, click **Save** in the upper right corner to complete the template creation.

g. (Optional) If you do not need to modify the template, click **Create Job** in the upper right corner to create a job based on the current template. For details about how to create a job, see **Creating a Flink SQL Job**, and **Creating a Flink Jar Job**.

- Creating a template based on an existing job template

  a. In the left navigation pane of the DLI management console, choose **Job Templates** > **Flink Templates**. Click the **Custom Templates** tab.

  b. In the row where the desired template is located in the custom template list, click **Edit** under **Operation** to enter the **Edit** page.

  c. After the modification is complete, set **Saving Mode** to **Save as New**.

  d. Click **Save** in the upper right corner to save the template as a new one.

- Creating a template using a created job

  a. In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

  b. Click **Create Job** in the upper right corner. The **Create Job** page is displayed.

  c. Specify parameters as required.

  d. Click **OK** to enter the editing page.

  e. After the SQL statement is compiled, click **Set as Template**.

  f. In the **Set as Template** dialog box that is displayed, specify **Name** and **Description** and click **OK**.

- Creating a template based on the existing job

  a. In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

  b. In the job list, locate the row where the job that you want to set as a template resides, and click **Edit** in the **Operation** column.

c. After the SQL statement is compiled, click **Set as Template**.

d. In the **Set as Template** dialog box that is displayed, specify **Name** and **Description** and click **OK**.

Creating a Job Based on a Template

You can create jobs based on sample templates or custom templates.

1. In the left navigation pane of the DLI management console, choose **Job Templates** > **Flink Templates**.
2. In the sample template list, click **Create Job** in the **Operation** column of the target template. For details about how to create a job, see **Creating a Flink SQL Job** and **Creating a Flink Jar Job**.

Modifying a Template

After creating a custom template, you can modify it as required. The sample template cannot be modified, but you can view the template details.

1. In the left navigation pane of the DLI management console, choose **Job Templates** > **Flink Templates**. Click the **Custom Templates** tab.
2. In the row where the template you want to modify is located in the custom template list, click **Edit** in the **Operation** column to enter the **Edit** page.
3. In the SQL statement editing area, modify the SQL statements as required.
4. Set **Saving Mode** to **Save Here**.
5. Click **Save** in the upper right corner to save the modification.

Deleting a Template

You can delete a custom template as required. The sample templates cannot be deleted. Deleted templates cannot be restored. Exercise caution when performing this operation.

1. In the left navigation pane of the DLI management console, choose **Job Templates** > **Flink Templates**. Click the **Custom Templates** tab.
2. In the custom template list, select the templates you want to delete and click **Delete** in the upper left of the custom template list.

   Alternatively, you can delete a template by performing the following operations: In the custom template list, locate the row where the template you want to delete resides, and click **More** > **Delete** in the **Operation** column.
3. In the displayed dialog box, click **Yes**.

8.3 Managing Spark SQL Templates

Scenario

You can modify a sample template to meet the Spark job requirements, saving time for editing SQL statements.

Currently, the cloud platform does not provide preset Spark templates. You can customize Spark job templates. This section describes how to create a Spark job template.

Creating a Management Job Template

To create a Spark job template, you can save a Spark job information as a template.

1. In the left navigation pane of the DLI management console, choose **Job Templates** > **Spark Templates**. The **Spark Templates** page is displayed.
2. Configure job parameters by referring to **Creating a Spark Job**.
3. After you finish editing the job, click **Save as Template**.
4. Enter the template name and other information as you need.
5. Set a template group for future management.
6. Click **OK**.

8.4 Appendix

8.4.1 TPC-H Sample Data in the SQL Template

TPC-H Sample Data

TPC-H is a test set developed by the Transaction Processing Performance Council (TPC) to simulate decision-making support applications. It is widely used in academia and industry to evaluate the performance of decision-making support technology. This business test has higher requirements on vendors, because it can comprehensively evaluate the overall business computing capability. With universal business significance, is widely used in analysis of bank credit, credit card, telecom operation, tax, as well as tobacco industry decision-making analysis.

The TPC-H benchmark test is developed from TPC-D (a standard specified by TPC in 1994 and used as the test benchmark for decision-making support systems). TPC-H implements a 3NF data warehouse that contains eight basic relationships, with a data volume range from 1 GB to 3 TB. The TPC-H benchmark test includes 22 queries (Q1 to Q22). The main evaluation indicator is the response time of each query (from submission to result return). The unit of the TPC-H benchmark test is the query number per hour (QphH@size). **H** indicates the average number of complex queries per hour. **size** indicates the size of database, which reflects the query processing capability of the system. TPC-H can evaluate key performance parameters that other tests cannot evaluate, because it is modeled based on the actual production and operation environment. In a word, the TPC-H standard by TPC meets the test requirements of data warehouse and motivate vendors and research institutes to stretch the limit of this technology.

In this example, DLI directly queries the TPC-H dataset on OBS. DLI has generated a standard TPC-H-2.18 dataset of 100 MB which is uploaded to the tpch folder on OBS. The read-only permission is granted to you to facilitate query operations.

TPC-H Test and Metrics

TPC-H test is divided into three sub-tests: data loading test, Power test, and Throughput test. Data loading indicates the process of setting up a test database, and the loading test is to test the data loading ability of DBMS. The first test is data loading test that tests data loading time, which is time-consuming. The second test is Power test, also called raw query. After data loading test is complete, the database is in the initial state without any other operation, especially the data in the buffer is not tested. Power test requires that the 22 queries be executed once in sequence and a pair of RF1 and RF2 operations be executed at the same time. The third test is Throughput test, the core and most complex test, more similar to the actual application environment. With multiple query statement groups and a pair of RF1 and RF2 update flows, Throughput test pose greater pressure on the SUT system than Power test does.

The basic data in the test is related to the execution time (the time of each data loading step, each query execution, and each update execution), based on which you can calculate the data loading time, Power@Size, Throughput@Size, qphH@Size and $/QphH@Size.

Power@Size is the result of the Power test, which is defined as the reciprocal of the geometric average value of the query time and change time. The formula is as follows:

$$\text{TPC-H Power@Size} = \frac{3600 * SF}{\sqrt[24]{\prod_{i=1}^{i=22} QI(i,0) * \prod_{j=1}^{j=2} RI(j,0)}}$$

Size indicates the data size. SF is the scaling factor of data scale. QI (i, 0) indicates the time of the ith query, in seconds. R (I j, 0) is the update time of RFj, in seconds.

Throughput@Size is the Throughput test result, which is defined as the reciprocal of the average value of all query execution time. The formula is as follows:

$$\text{QphH@Size} = \sqrt{Power\ @\ Size * Throughput\ @\ Size}$$

Service Scenario

You can use the built-in TPC-H test suite of DLI to perform interactive query without uploading data.

Advantages of DLI Built-in TPC-H

- You can log in to DLI and get permission to run SQL statements without creating tables or import data.
- The 22 preset TPC-H SQL query templates with rich functions meet the requirements of most business scenarios. You do not need to download TPC-H query statements, which saves your time and energy.

- Data Lake gives you brand-new experience of serverless DLI product within the minimum time.

Precautions

When a sub-account uses the TPC-H test suite, the main account needs to grant the sub-account the OBS access permission and the permission to view the main account table. If the master account has not logged in to DLI, the sub-account needs to have the permissions to create databases and tables in addition to the preceding permissions.

Operation Description

For details, see **Managing SQL Templates**.

9 Enhanced Datasource Connections

9.1 Overview

What Is Enhanced Datasource Connection?

Typically, you cannot use DLI to directly access a data source in a VPC other than the one where DLI is because the network between DLI and the data source is disconnected. For proper access, you need to establish a network connection between them.

DLI provides enhanced connections. Establishing a VPC peering connection allows DLI to communicate with the VPC of the data source, supporting cross-source data analysis.

For details about the data sources that support cross-source access, see **Cross-Source Analysis Development Methods**.

Constraints

- Datasource connections cannot be created for the **default** queue.
- Flink jobs can directly access DIS, OBS, and SMN data sources without using datasource connections.
- **VPC Administrator** permissions are required for enhanced connections to use VPCs, subnets, routes, VPC peering connections.
- If you use an enhanced datasource connection, the CIDR block of the elastic resource pool or queue cannot overlap with that of the data source.
- Only queues bound with datasource connections can access datasource tables.
- Datasource tables do not support the preview function.
- When checking the connectivity of datasource connections, the constraints on IP addresses are as follows:
  - The IP address must be valid, which consists of four decimal numbers separated by periods (.). The value ranges from 0 to 255.
  - During the test, you can add a port after the IP address and separate them with colons (:). The port can contain a maximum of five digits. The value ranges from 0 to 65535.

For example, **192.168.***xx.xx* or **192.168.***xx.xx***:8181**.

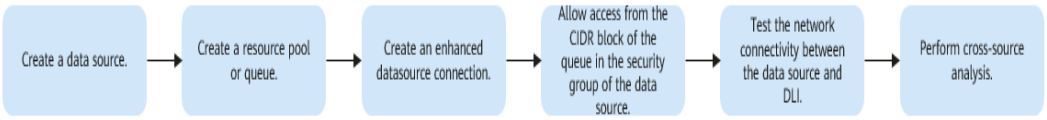
- When checking the connectivity of datasource connections, the constraints on domain names are as follows:
  - The domain name can contain 1 to 255 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
  - The top-level domain name must contain at least two letters, for example, **.com**, **.net**, and **.cn**.
  - During the test, you can add a port after the domain name and separate them with colons (:). The port can contain a maximum of five digits. The value ranges from 0 to 65535.

    For example, **example.com:8080**.

Cross-Source Analysis Process

To use DLI for cross-source analysis, you need to create a datasource connection to connect DLI to the data source, and then develop jobs to access the data source.

Figure 9-1 Cross-source analysis flowchart



9.2 Cross-Source Analysis Development Methods

Cross-Source Analysis

If DLI needs to access external data sources, you need to establish enhanced datasource connections to enable the network between DLI and the data sources, and then develop different types of jobs to access the data sources. This is the process of DLI cross-source analysis.

This section describes how to develop data sources supported by DLI for cross-source analysis.

Notes

- Flink jobs can directly access DIS, OBS, and SMN data sources without using datasource connections.
- You are advised to use enhanced datasource connections to connect DLI to data sources.

DLI Supported Data Sources

Table 9-1 lists the data sources supported by DLI. For details about how to use the data sources, see *Data Lake Insight SQL Syntax Reference*.

Table 9-1 Supported data sources

Service	Spark SQL Job	Spark Jar Job	Flink SQL Job	Flink Jar Job
APIG	x	x	√	x
CSS	√	√	√	√
DCS Redis	√	√	√	√
DDS Mongo	√	√	√	√
DMS Kafka	x	x	√	√
GaussDB(DWS)	√	√	√	√
MRS HBase	√	√	√	√
MRS Kafka	x	x	√	√
MRS OpenTSDB	√	√	x	√
RDS MySQL	√	√	√	√
RDS PostGre	√	√	√	√

9.3 Creating an Enhanced Datasource Connection

Scenario

Create an enhanced datasource connection for DLI to access, import, query, and analyze data of other data sources.

For example, to connect DLI to the MRS, RDS, CSS, Kafka, or GaussDB(DWS) data source, you need to enable the network between DLI and the VPC of the data source.

Create an enhanced datasource connection on the console.

Constraints

- Datasource connections cannot be created for the **default** queue.
- Flink jobs can directly access DIS, OBS, and SMN data sources without using datasource connections.
- **VPC Administrator** permissions are required for enhanced connections to use VPCs, subnets, routes, VPC peering connections.
- If you use an enhanced datasource connection, the CIDR block of the elastic resource pool or queue cannot overlap with that of the data source.
- Only queues bound with datasource connections can access datasource tables.
- Datasource tables do not support the preview function.
- When checking the connectivity of datasource connections, the constraints on IP addresses are as follows:

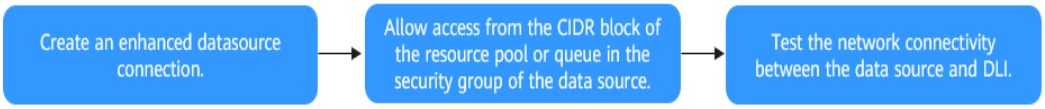
  - The IP address must be valid, which consists of four decimal numbers separated by periods (.). The value ranges from 0 to 255.
  - During the test, you can add a port after the IP address and separate them with colons (:). The port can contain a maximum of five digits. The value ranges from 0 to 65535.

    For example, **192.168.***xx.xx* or **192.168.***xx.xx***:8181**.
- When checking the connectivity of datasource connections, the constraints on domain names are as follows:
  - The domain name can contain 1 to 255 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
  - The top-level domain name must contain at least two letters, for example, **.com**, **.net**, and **.cn**.
  - During the test, you can add a port after the domain name and separate them with colons (:). The port can contain a maximum of five digits. The value ranges from 0 to 65535.

    For example, **example.com:8080**.

Process

Figure 9-2 Enhanced datasource connection creation flowchart



Prerequisites

- An elastic resource pool or queue has been created.
- You have obtained the VPC, subnet, private IP address, port, and security group information of the external data source.
- The security group of the external data source has allowed access from the CIDR block of the elastic resource pool or queue.

Procedure

Step 1 Create an Enhanced Datasource Connection

1. Log in to the DLI management console.
2. In the left navigation pane, choose **Datasource Connections**.
3. On the **Enhanced** tab page displayed, click **Create**.

   Configure parameters according to **Table 9-2**.

Table 9-2 Parameters

Parameter	Description
Connection Name	Name of the created datasource connection. – The name can contain only letters, digits, and underscores (_). The parameter must be specified. – A maximum of 64 characters are allowed.
VPC	VPC used by the data source.
Subnet	Subnet used by the data source.
Host Information	In this text field, you can configure the mapping between host IP addresses and domain names so that jobs can only use the configured domain names to access corresponding hosts. This parameter is optional. For example, when accessing the HBase cluster of MRS, you need to configure the host name (domain name) and IP address of the ZooKeeper instance. Enter one record in each line in the format of *IP address Host name*/*Domain name*. Example: 192.168.0.22 node-masterxxx1.com 192.168.0.23 node-masterxxx2.com For details about how to obtain host information, see **How Do I Obtain MRS Host Information?**.
Tags	Tags used to identify cloud resources. A tag includes the tag key and tag value. If you want to use the same tag to identify multiple cloud resources, that is, to select the same tag from the drop-down list box for all services, you are advised to create predefined tags on the Tag Management Service (TMS). **NOTE** – A maximum of 20 tags can be added. – Only one tag value can be added to a tag key. – The key name in each resource must be unique. – Tag key: Enter a tag key name in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with **_sys_**. – Tag value: Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 225 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

4. Click **OK**.

After the creation is complete, the enhanced datasource connection is in the **Active** state, indicating that the connection is successfully created.

Step 2 Security Group Where the Data Source Belongs Allows Access from the CIDR Block of the Elastic Resource Pool

1. On the DLI management console, obtain the network segment of the elastic resource pool or queue.

   Choose **Resources** > **Queue Management** from the left navigation pane. On the page displayed, locate the queue on which jobs are running, and click the button next to the queue name to obtain the CIDR block of the queue.

2. Log in to the VPC console and find the VPC the data source belongs to.
3. On the network console, choose **Virtual Private Cloud** > **Network Interfaces**. On the **Network Interfaces** tab page displayed, search for the security group name, click **More** in the **Operation** column, and select **Change Security Group**.
4. In the navigation pane on the left, choose **Access Control** > **Security Groups**.
5. Click the name of the security group to which the external data source belongs.
6. Click the **Inbound Rules** tab and add a rule to allow access from the CIDR block of the queue.

   Configure the inbound rule parameters according to **Table 9-3**.

Table 9-3 Inbound rule parameters

Parameter	Description	Example Value
Priority	Priority of a security group rule. The priority value ranges from 1 to 100. The default value is **1**, indicating the highest priority. A smaller value indicates a higher priority of a security group rule.	1
Action	Action of the security group rule.	Allow
Protocol & Port	– Network protocol. The value can be **All**, **TCP**, **UDP**, **ICMP**, or **GRE**. – Port: Port or port range over which the traffic can reach your instance. The port ranges from 1 to 65535.	In this example, select **TCP**. Leave the port blank or set it to the data source port.
Type	Type of IP addresses.	IPV4

Parameter	Description	Example Value
Source	Allows access from IP addresses or instances in another security group.	In this example, enter the obtained queue CIDR block.
Description	Supplementary information about the security group rule. This parameter is optional.	_

Step 3 Test the Connectivity Between the DLI Queue and the Data Source

1. Obtain the private IP address and port number of the data source.

   Take the RDS data source as an example. On the **Instances** page, click the target DB instance. On the page displayed, locate the **Connection Information** pane and view the private IP address. In the **Connection Information** pane, locate the **Database Port** to view the port number of the RDS DB instance.
2. In the navigation pane of the DLI management console, choose **Resources** > **Queue Management**.
3. Locate the queue bound with the enhanced datasource connection, click **More** in the **Operation** column, and select **Test Address Connectivity**.
4. Enter the data source connection address and port number to test the network connectivity.

   Format: *IP address*:*Port number*

> **CAUTION**
>
> Before testing the connection, ensure that the security group of the external data source has allowed access from the CIDR block of the queue.

----End

9.4 Deleting an Enhanced Datasource Connection

Scenario

Delete an enhanced datasource connection that is no longer used on the console.

Procedure

1. Log in to the DLI management console.
2. In the left navigation pane, choose **Datasource Connections**.
3. On the **Enhanced** tab page displayed, locate the enhanced datasource connection to be deleted and click **Delete** in the **Operation** column.

4. In the dialog box displayed, click **Yes**.

9.5 Modifying Host Information

Scenario

Host information is the mapping between host IP addresses and domain names. After you configure host information, jobs can only use the configured domain names to access corresponding hosts. After a datasource connection is created, you can modify the host information.

When accessing the HBase cluster of MRS, you need to configure the host name (domain name) and IP address of the instance.

Constraints

You have obtained the MRS host information by referring to **How Do I Obtain MRS Host Information?**

Modifying Host Information

1. Log in to the DLI management console.
2. In the left navigation pane, choose **Datasource Connections**.
3. On the **Enhanced** tab page displayed, locate the enhanced datasource connection to be modified, click **More** in the **Operation** column, and select **Modify Host**.
4. In the **Modify Host** dialog box displayed, enter the obtained host information.

   Enter host information in the format of *Host IP address Host name*. Information about multiple hosts is separated by line breaks.

   Example:

   192.168.0.22 node-masterxxx1.com

   192.168.0.23 node-masterxxx2.com

   Obtain the MRS host information by referring to **How Do I Obtain MRS Host Information?**
5. Click **OK**.

How Do I Obtain MRS Host Information?

- **Method 1: View MRS host information on the management console.**

  To obtain the host name and IP address of an MRS cluster, for example, MRS 3.*x*, perform the following operations:

  a. Log in to the MRS management console.
  b. On the **Active Clusters** page displayed, click your desired cluster to access its details page.
  c. Click the **Components** tab.
  d. Click **ZooKeeper**.
  e. Click the **Instance** tab to view corresponding service IP addresses. You can select any service IP address.

f. Modify host information by referring to **Modifying Host Information**.

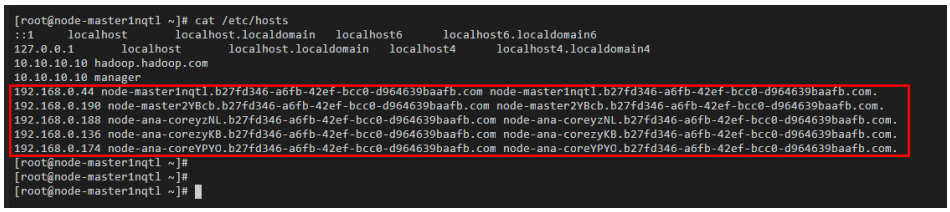
NOTE

If the MRS cluster has multiple IP addresses, enter any service IP address when creating a datasource connection.

- **Method 2: Obtain MRS host information from the /etc/hosts file on an MRS node.**
  a. Log in to any MRS node as user **root**.
  b. Run the following command to obtain MRS hosts information. Copy and save the information.

     cat /etc/hosts

     Figure 9-3 Obtaining hosts information

  c. Modify host information by referring to **Modifying Host Information**.
- **Method 3: Log in to FusionInsight Manager to obtain host information.**
  a. Log in to FusionInsight Manager.
  b. On FusionInsight Manager, click **Hosts**. On the **Hosts** page, obtain the host names and service IP addresses of the MRS hosts.
  c. Modify host information by referring to **Modifying Host Information**.

9.6 Binding an Elastic Resource Pool

Scenario

To connect other resource pools to data sources through enhanced datasource connections, bind enhanced datasource connections to resource pools on the **Enhanced** tab page.

Constraints

- The CIDR block of the DLI queue to be bound with a datasource connection cannot overlap with that of the data source.
- The **default** queue preset in the system cannot be bound with a datasource connection.

Procedure

1. Log in to the DLI management console.
2. In the left navigation pane, choose **Datasource Connections**.
3. On the **Enhanced** tab page displayed, bind an enhanced datasource connection to an elastic resource pool:

a. Locate your desired enhanced datasource connection, click **More** in the **Operation** column, and select **Bind Resource Pool**.

b. In the **Bind Resource Pool** dialog box, select the resource pool to be bound for **Resource Pool**.

c. Click **OK**.

4. View the connection status on the **Enhanced** tab page.

   - After an enhanced datasource connection is created, the status is **Active**, but it does not indicate that the queue is connected to the data source. Go to the queue management page to check whether the data source is connected. The procedure is as follows:

     i. In the navigation pane on the left, choose **Resources** > **Queue Management**. On the page displayed, locate a desired queue.

     ii. Click **More** in the **Operation** column and select **Test Address Connectivity**.

     iii. Enter the IP address and port number of the data source.

   - On the details page of an enhanced datasource connection, you can view information about the VPC peering connection.

     - VPC peering ID: ID of the VPC peering connection created in the cluster to which the queue belongs.

       A VPC peering connection is created for each queue bound to an enhanced datasource connection. The VPC peering connection is used for cross-VPC communication. Ensure that the security group used by the data source allows access from the CIDR block of the DLI queue, and do not delete the VPC peering connection during the datasource connection.

     - Status of the VPC peering connection:

       The status of a datasource connection can be **Creating**, **Active**, or **Failed**.

       If the connection status is **Failed**, click on the left to view the detailed error information.

9.7 Unbinding an Elastic Resource Pool

Scenario

Unbind an enhanced datasource connection from an elastic resource pool that does not need to access a data source through an enhanced datasoruce connection.

Constraints

If the status of the VPC peering connection created for binding an enhanced datasource connection to an elastic resource pool is **Failed**, the elastic resource pool cannot be unbound.

Procedure

1. Log in to the DLI management console.
2. In the left navigation pane, choose **Datasource Connections**.
3. On the **Enhanced** tab page displayed, use either of the following methods to unbind an enhanced datasource connection from an elastic resource pool:
   - Method 1:
     i. Locate your desired enhanced datasource connection, click **More** in the **Operation** column, and select **Unbind Resource Pool**.
     ii. In the **Unbind Resource Pool** dialog box, select the resource pool to be unbound for **Resource Pool**.
     iii. Click **OK**.
   - Method 2:
     i. Click your desired enhanced datasource connection in the list.
     ii. Locate your desired resource pool and click **Unbind Resource Pool** in the **Operation** column.
     iii. Click **OK**.

9.8 Adding a Route

Scenario

A route is configured with the destination, next hop type, and next hop to determine where the network traffic is directed. Routes are classified into system routes and custom routes.

After an enhanced connection is created, the subnet is automatically associated with the default route. You can add custom routes as needed to forward traffic destined for the destination to the specified next hop.

NOTE

- When an enhanced connection is created, the associated route table is the one associated with the subnet of the data source.
- The route to be added in the **Add Route** dialog box must be one in the route table associated with the subnet of the resource pool.
- The subnet of the data source must be different from that used by the resource pool. Otherwise, a network segment conflict occurs.

Procedure

1. Log in to the DLI management console.
2. In the left navigation pane, choose **Datasource Connections**.
3. On the **Enhanced** tab page displayed, locate the row containing the enhanced connection to which a route needs to be added, and add the route.
   - Method 1:
     i. On the **Enhanced** tab page displayed, locate the enhanced datasource connection to which a route needs to be added and click **Manage Route** in the **Operation** column.

ii. Click **Add Route**.

iii. In the **Add Route** dialog box, enter the route information. For details about the parameters, see **Table 9-4**.

iv. Click **OK**.

- Method 2:

i. On the **Enhanced** tab page displayed, locate the enhanced datasource connection to which a route needs to be added, click **More** in the **Operation** column, and select **Add Route**.

ii. In the **Add Route** dialog box, enter the route information. For details about the parameters, see **Table 9-4**.

iii. Click **OK**.

Table 9-4 Parameters for adding a custom route

Parameter	Description
Route Name	Name of a custom route, which is unique in the same enhanced datasource scenario. The name can contain 1 to 64 characters. Only digits, letters, underscores (_), and hyphens (-) are allowed.
IP Address	Custom route CIDR block. The CIDR block of different routes can overlap but cannot be the same.

4. After adding a route, you can view the route information on the route details page.

9.9 Deleting a Route

Scenario

Delete a route that is no longer used.

Constraints

A custom route table cannot be deleted if it is associated with a subnet.

Procedure

1. Log in to the DLI management console.
2. In the left navigation pane, choose **Datasource Connections**.
3. On the **Enhanced** tab page displayed, locate the row containing the enhanced connection from which the route needs to be deleted, and delete the route.
   - Method 1:

     i. On the **Enhanced** tab page displayed, locate the enhanced connection from which the route needs to be deleted and click **Manage Route** in the **Operation** column.

ii. Locate the route to be deleted and click **Delete** in the **Operation** column.

iii. In the dialog box displayed, click **OK**.

– Method 2:

i. On the **Enhanced** tab page displayed, locate the enhanced connection from which the route needs to be deleted, click **More** in the **Operation** column, and select **Delete Route**.

ii. In the **Delete Route** dialog box displayed, confirm the route information.

iii. Click **Yes**.

9.10 Enhanced Connection Permission Management

Scenario

Enhanced connections support user authorization by project. After authorization, users in the project have the permission to perform operations on the enhanced connection, including viewing the enhanced connection, binding a created resource pool to the enhanced connection, and creating custom routes. In this way, the enhanced connection can be used across projects. Grant and revoke permissions to and from a user for an enhanced connection.

NOTE

- If the authorized projects belong to different users in the same region, you can use the user account of the authorized projects to log in.
- If the authorized projects belong to the same user in the same region, you can use the current account to switch to the corresponding project.

Use Cases

Project B needs to access the data source of project A. The operations are as follows:

- For Project A:
  a. Log in to DLI using the account of project A.
  b. Create an enhanced datasource connection **ds** in DLI based on the VPC information of the corresponding data source.
  c. Grant project B the permission to access the enhanced datasource connection **ds**.
- For Project B:
  a. Log in to DLI using the account of project B.
  b. Bind the enhanced datasource connection **ds** to a queue.
  c. (Optional) Set host information and create a route.

After creating a VPC peering connection and route between the enhanced datasource connection of project A and the queue of project B, you can create a job in the queue of project B to access the data source of project A.

Procedure

1. Log in to the DLI management console.
2. In the left navigation pane, choose **Datasource Connections**.
3. On the **Enhanced** tab page displayed, locate the desired enhanced connection, click **More** in the **Operation** column, and select **Manage Permission**.
   - **Granting permission**
     i. In the **Permissions** dialog box displayed, select **Grant Permission** for **Set Permission**.
     ii. Enter the project ID.
     iii. Click **OK** to grant the resource pool operation permission to the project.
   - **Revoking permission**
     i. In the **Permissions** dialog box displayed, select **Revoke Permission** for **Set Permission**.
     ii. Select a project ID.
     iii. Click **OK** to revoke the resource pool operation permission from the specified project.

9.11 Enhanced Datasource Connection Tag Management

Scenario

A tag is a key-value pair customized by users and used to identify cloud resources. It helps users to classify and search for cloud resources. A tag consists of a tag key and a tag value.

If you use tags in other cloud services, you are advised to create the same tag key-value pairs for cloud resources used by the same business to keep consistency.

DLI supports the following two types of tags:

- Resource tags: non-global tags created on DLI.
- Predefined tags: global tags created on Tag Management Service (TMS).

DLI allows you to add, modify, or delete tags for datasource connections.

Procedure

1. In the left navigation pane of the DLI management console, choose **Datasource Connections**.
2. In the **Operation** column of the link, choose **More** > **Tags**.
3. The tag management page is displayed, showing the tag information about the current connection.
4. Click **Add/Edit Tag**. The **Add/Edit Tag** dialog is displayed. Add or edit tag keys and values and click **OK**.

Table 9-5 Tag parameters

Parameter	Description
Tag key	You can perform the following operations: • Click the text box and select a predefined tag key from the drop-down list. To add a predefined tag, you need to create one on TMS and then select it from the **Tag key** drop-down list. You can click **View predefined tags** to go to the **Predefined Tags** page of the TMS console. Then, click **Create Tag** in the upper corner of the page to create a predefined tag. • Enter a tag key in the text box. **NOTE** A tag key can contain a maximum of 128 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed, but the value cannot start or end with a space or start with **_sys_**.
Tag value	You can perform the following operations: • Click the text box and select a predefined tag value from the drop-down list. • Enter a tag value in the text box. **NOTE** A tag value can contain a maximum of 225 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed. The value cannot start or end with a space.

5. Click **OK**.
6. (Optional) To delete a tag, locate the row where the tag locates in the tag list and click **Delete** in the **Operation** column to delete the tag.

10 Datasource Authentication

10.1 Introduction

What Is Datasource Authentication?

Datasource authentication is used to manage authentication information for accessing specified data sources. After datasource authentication is configured, you do not need to repeatedly configure data source authentication information in jobs, improving data source authentication security while enabling DLI to securely access data sources.

Constraints

- Compared with datasource authentication provided by DLI, you are advised to use Data Encryption Worksop (DEW) to store data source authentication information.
- Only Spark SQL and Flink OpenSource SQL 1.12 jobs support datasource authentication.
- The version of the cluster where the queue belongs may not support datasource authentication. You are advised to create a queue to run Flink jobs.
- DLI supports four types of datasource authentication. Select an authentication type specific to each data source.
  - CSS: applies to 6.5.4 or later CSS clusters with the security mode enabled.
  - Kerberos: applies to MRS security clusters with Kerberos authentication enabled.
  - Kafka_SSL: applies to Kafka with SSL enabled.
  - Password: applies to GaussDB(DWS), RDS, DDS, and DCS.

Datasource Authentication Types

DLI supports four types of datasource authentication. Select an authentication type specific to each data source.

- CSS: applies to 6.5.4 or later CSS clusters with the security mode enabled. During the configuration, you need to specify the username, password, and authentication certificate of the cluster and store the information in DLI through datasource authentication so that DLI can securely access CSS data sources. For details, see **Creating a CSS Datasource Authentication**.
- Kerberos: applies to MRS security clusters with Kerberos authentication enabled. During the configuration, you need to specify MRS cluster authentication credentials, including the **krb5.conf** and **user.keytab** files. For details, see **Creating a Kerberos Datasource Authentication**.
- Kafka_SSL: applies to Kafka with SSL enabled. During the configuration, you need to specify the KafkaTruststore path and password. For details, see **Creating a Kafka_SSL Datasource Authentication**.
- Password: applies to GaussDB(DWS), RDS, DDS, and DCS data sources. During the configuration, you need to store the passwords of the data sources in DLI. For details, see **Creating a Password Datasource Authentication**.

Jobs That Can Connect to Data Sources Through Datasource Authentication

Different types of jobs can connect to data sources through different types of datasource authentication.

- For details about the data sources that Spark SQL jobs can connect to through datasource authentication and their constraints, see **Table 10-1**.
- For details about the data sources that Flink OpenSource SQL 1.12 jobs can connect to through datasource authentication and their constraints, see **Table 10-2**.

Table 10-1 Data sources that Spark SQL jobs can connect to through datasource authentication

Datasource Authentication Type	Data Source	Constraints
CSS	CSS	The CSS cluster version must be 6.5.4 or later. The security mode has been enabled for the CSS cluster.
Password	GaussDB(DWS), RDS, DDS, and Redis	-

Table 10-2 Data sources that Flink OpenSource SQL 1.12 jobs can connect to through datasource authentication

Table Type	Datasource Authentication Type	Data Source	Constraints
Source table	Kerberos	HBase	Kerberos authentication has been enabled for the MRS cluster.
		Kafka	Kerberos authentication has been enabled for MRS Kafka.
	Kafka_SSL	Kafka	SASL_SSL authentication has been enabled for DMS Kafka. SASL authentication has been enabled for MRS Kafka. SSL authentication has been enabled for MRS Kafka.
	Password	GaussDB(DWS), RDS, and Redis	-
Result table	Kerberos	HBase	Kerberos authentication has been enabled for the MRS cluster.
		Kafka	Kerberos authentication has been enabled for MRS Kafka.
	Kafka_SSL	Kafka	SASL_SSL authentication has been enabled for DMS Kafka. SASL authentication has been enabled for MRS Kafka. SSL authentication has been enabled for MRS Kafka.
	Password	GaussDB(DWS), RDS, CSS, and Redis	-
Dimension table	Kerberos	HBase	Kerberos authentication has been enabled for the MRS cluster.
	Password	GaussDB(DWS), RDS, and Redis	-

10.2 Creating a CSS Datasource Authentication

Scenario

Create a CSS datasource authentication on the DLI console to store the authentication information of the CSS security cluster to DLI. This will allow you to

access to the CSS security cluster without having to configure a username and password in SQL jobs.

Create a datasource authentication for a CSS security cluster on the DLI console.

Notes

A CSS security cluster has been created and has met the following conditions:

- The cluster version is 6.5.4 or later.
- The security mode has been enabled for the cluster.

Procedure

1. Download the authentication credential of the CSS security cluster.
   a. Log in to the CSS management console and choose **Clusters** > **Elasticsearch**.
   b. On the **Clusters** page displayed, click the cluster name.
   c. On the **Cluster Information** page displayed, find the security mode and download the certificate of the CSS security cluster.
2. Upload the authentication credential to the OBS bucket.
3. Create a datasource authentication.
   a. Log in to the DLI management console.
   b. Choose **Datasource Connections**. On the page displayed, click **Datasource Authentication**.
   c. Click **Create**.

      Configure CSS authentication parameters according to **Table 10-3**.

Table 10-3 Parameters

Parameter	Description
Authentication Certificate	Name of the datasource authentication information to be created. • The name can contain only digits, letters, and underscores (_), but cannot contain only digits or start with an underscore (_). • The length of the database name cannot exceed 128 characters. • It is recommended that the name contain the CSS security cluster name to distinguish security authentication information of different clusters.
Type	Select **CSS**.
Username	Username for logging in to the security cluster.
Password	The password of the security cluster

Parameter	Description
Certificate Path	Enter the OBS path to which the security certificate is uploaded, that is, the OBS bucket address in 2.

4. Create a table to access the CSS cluster.

   When creating a table, associate the table with the created datasource authentication to access the CSS cluster.

   For example, when using Spark SQL to create a table for accessing the CSS cluster, configure **es.certificate.name** to set the datasource authentication name and then connect to the CSS security cluster.

10.3 Creating a Kerberos Datasource Authentication

Scenario

Create a Kerberos datasource authentication on the DLI console to store the authentication information of the data source to DLI. This will allow you to access to the data source without having to configure a username and password in SQL jobs.

NOTE

- When Kerberos authentication is enabled for MRS Kafka but SSL authentication is disabled, create a Kerberos authentication. When creating a table, configure **krb_auth_name** to associate the datasource authentication.
- If Kerberos authentication and SSL authentication are both enabled for MRS Kafka, you need to create Kerberos and Kafka_SSL authentications. When creating a table, configure **krb_auth_name** and **ssl_auth_name** to associate the datasource authentications.
- Datasource authentication is not required when Kerberos authentication is disabled but SASL authentication is enabled for MRS Kafka (for example, when a username and a password are used for PlainLoginModule authentication).
- When Kerberos authentication is disabled but SSL authentication is enabled for MRS Kafka, you need to create a Kafka_SSL authentication. When creating a table, configure **ssl_auth_name** to associate the datasource authentication.
- When Kerberos authentication is disabled but SASL authentication and SSL authentication are enabled for MRS Kafka, you need to create a Kafka_SSL authentication. When creating a table, configure **ssl_auth_name** to associate the datasource authentication.

Data Sources Supported by Kerberos Datasource Authentication

Table 10-4 lists the data sources supported by Kerberos datasource authentication.

Table 10-4 Data sources supported by Kerberos datasource authentication

Job Type	Table Type	Data Source	Constraints
Flink OpenSource SQL	Source table	HBase	Kerberos authentication has been enabled for the MRS cluster.
		Kafka	Kerberos authentication has been enabled for MRS Kafka.
	Result table	HBase	Kerberos authentication has been enabled for the MRS cluster.
		Kafka	Kerberos authentication has been enabled for MRS Kafka.
	Dimension table	HBase	Kerberos authentication has been enabled for the MRS cluster.

Procedure

1. Download the authentication credential of the data source.
   a. Log in to MRS Manager.
   b. Choose **System** > **Permission** > **User**.
   c. Click **More**, select **Download Authentication Credential**, save the file, and decompress it to obtain the **keytab** and **krb5.conf** files.
2. Upload the authentication credential to the OBS bucket.
3. Create a datasource authentication.
   a. Log in to the DLI management console.
   b. Choose **Datasource Connections**. On the page displayed, click **Datasource Authentication**.
   c. Click **Create**.
   Configure Kerberos authentication parameters according to **Table 10-5**.

   Table 10-5 Parameters

Parameter	Description
Type	Select **Kerberos**.

Parameter	Description
Authentication Certificate	Name of the datasource authentication to be created. • The name can contain only digits, letters, and underscores (_), but cannot contain only digits or start with an underscore (_). • The name can contain a maximum of 128 characters. • It is recommended that the name contain the MRS security cluster name to distinguish security authentication information of different clusters.
Username	Username for logging in to the security cluster.
krb5_conf Path	OBS path to which the **krb5.conf** file is uploaded. **NOTE** The **renew_lifetime** configuration item under **[libdefaults]** must be removed from **krb5.conf**. Otherwise, the "Message stream modified (41)" error may occur.
keytab Path	OBS path to which the **user.keytab** file is uploaded.

4. Create a table to access the MRS cluster.

   When creating a data source, associate the data source with the created datasource authentication to access the data source.

   Table 10-6 lists the fields used to associate with the datasource authentication during table creation.

   Table 10-6 Fields that are used to associate with Kerberos datasource authentication during table creation

Job Type	Data Source	Parameter	Mandatory	Data Type	Description
Flink OpenSource SQL	HBase	krb_auth_name	No	String	This field is used to associate datasource authentications when source, result, and dimension tables are created.

Job Type	Data Source	Parameter	Mandatory	Data Type	Description
	Kafka	krb_auth_name	No	String	This field is used to associate datasource authentications when source and result tables are created. Name of the created Kerberos datasource authentication. If SASL_PLAINTEXT and Kerberos authentication are both used, you need to configure the following parameters: ● 'properties.sasl.mechanism' = 'GSSAPI' ● 'properties.security.protocol' = 'SASL_PLAINTEXT'

For details about how to create a table, see *Data Lake Insight Syntax Reference*.

10.4 Creating a Kafka_SSL Datasource Authentication

Scenario

Create a Kafka_SSL datasource authentication on the DLI console to store the Kafka authentication information to DLI. This will allow you to access to Kafka instances without having to configure a username and password in SQL jobs.

NOTE

- When Kerberos authentication is enabled for MRS Kafka but SSL authentication is disabled, create a Kerberos authentication. When creating a table, configure **krb_auth_name** to associate the datasource authentication.
- If Kerberos authentication and SSL authentication are both enabled for MRS Kafka, you need to create Kerberos and Kafka_SSL authentications. When creating a table, configure **krb_auth_name** and **ssl_auth_name** to associate the datasource authentications.
- Datasource authentication is not required when Kerberos authentication is disabled but SASL authentication is enabled for MRS Kafka (for example, when a username and a password are used for PlainLoginModule authentication).
- When Kerberos authentication is disabled but SSL authentication is enabled for MRS Kafka, you need to create a Kafka_SSL authentication. When creating a table, configure **ssl_auth_name** to associate the datasource authentication.
- When Kerberos authentication is disabled but SASL authentication and SSL authentication are enabled for MRS Kafka, you need to create a Kafka_SSL authentication. When creating a table, configure **ssl_auth_name** to associate the datasource authentication.

Data Sources Supported by Kafka_SSL Datasource Authentication

Table 10-7 lists the data sources supported by Kafka_SSL datasource authentication.

Table 10-7 Data sources supported by Kafka_SSL datasource authentication

Job Type	Table Type	Data Source	Constraints
Flink OpenSource SQL	Source table and result table	Kafka	SASL_SSL authentication has been enabled for DMS Kafka. SASL authentication has been enabled for MRS Kafka. SSL authentication has been enabled for MRS Kafka.

Procedure

1. Download the authentication credential.
   - **DMS Kafka**
     i. Log in to the DMS (for Kafka) console and click a Kafka instance to access its details page.
     ii. In the connection information, find the SSL certificate and click **Download**.
     Decompress the downloaded **kafka-certs** package to obtain the **client.jks** and **phy_ca.crt** files.
   - **MRS Kafka**
     i. Log in to MRS Manager.
     ii. Choose **System** > **Permission** > **User**.
     iii. Click **More**, select **Download Authentication Credential**, save the file, and decompress it to obtain the truststore file.
2. Upload the authentication credential to the OBS bucket.
3. Create a datasource authentication.
   a. Log in to the DLI management console.
   b. Choose **Datasource Connections**. On the page displayed, click **Datasource Authentication**.
   c. Click **Create**.
   Configure Kafka authentication parameters according to Table 10-8.

Table 10-8 Parameters

Parameter	Description
Type	Select **Kafka_SSL**.

Parameter	Description
Authentication Certificate	Name of the datasource authentication to be created. • The name can contain only digits, letters, and underscores (_), but cannot contain only digits or start with an underscore (_). • The name can contain a maximum of 128 characters.
Truststore Path	OBS path to which the SSL truststore file is uploaded. • For MRS Kafka, enter the OBS path of the **Truststore.jks** file. • For DMS Kafka, enter the OBS path of the **client.jks** file.
Truststore Password	Truststore password.
Keystore Path	OBS path to which the SSL keystore file (key and certificate) is uploaded.
Keystore Password	Keystore (key and certificate) password.
Key Password	Password of the private key in the keystore file.

4. Access Kafka with SASL_SSL authentication enabled.

   When creating a data source, associate the data source with the created datasource authentication to access the data source.

   Table 10-9 lists the fields used to associate with the datasource authentication during table creation.

Table 10-9 Fields that are used to associate with Kafka_SSL datasource authentication during table creation

Parameter	Mandatory	Data Type	Description
ssl_auth_name	No	String	This field is used to associate datasource authentications when source, result, and dimension tables are created. Name of the created Kafka_SSL datasource authentication. This configuration is used when SSL is configured for Kafka. • If only SSL is used, configure the following parameter: 'properties.security.protocol '= 'SSL'; • If SASL_SSL is used, configure the following parameters: – 'properties.security.protocol' = 'SASL_SSL', – 'properties.sasl.mechanism' ='GSSAPI or PLAIN' – 'properties.sasl.jaas.config' = 'org.apache.kafka.common.security.plain.PlainLoginModule required username=\"xxx\" password=\"xxx\";'

For details about how to create a table, see *Data Lake Insight Syntax Reference*.

10.5 Creating a Password Datasource Authentication

Scenario

Create a password datasource authentication on the DLI console to store passwords of the GaussDB(DWS), RDS, DCS, and DDS data sources to DLI. This will allow you to access to the data sources without having to configure a username and password in SQL jobs.

Data Sources Supported by Password Datasource Authentication

Table 10-10 lists the data sources supported by password datasource authentication.

Table 10-10 Data sources supported by password datasource authentication

Job Type	Table Type	Data Source
Spark SQL	-	GaussDB(DWS), RDS, DDS, and Redis
Flink OpenSource SQL	Source table	GaussDB(DWS), RDS, and Redis
	Result table	GaussDB(DWS), RDS, CSS, and Redis
	Dimension table	GaussDB(DWS), RDS, and Redis

Procedure

1. Create a Kafka instance.

   When creating a Kafka instance, enable SASL_SSL for Kafka. Once SASL_SSL is enabled, data can be encrypted for transmission, which improves security.

2. Download the authentication credential.
   a. Log in to the Kafka console and click a Kafka instance to access its details page.
   b. In the connection information, find the SSL certificate and click **Download**.
3. Upload the authentication credential to the OBS bucket.
4. Create a datasource authentication.
   a. Log in to the DLI management console.
   b. Choose **Datasource Connections**. On the page displayed, click **Datasource Authentication**.
   c. Click **Create**.

      Configure authentication parameters according to **Table 10-11**.

      Table 10-11 Parameters

Parameter	Description
Type	Select **Password**.
Authentication Certificate	Name of the datasource authentication to be created. • The name can contain only digits, letters, and underscores (_), but cannot contain only digits or start with an underscore (_). • The name can contain a maximum of 128 characters.
Username	Username for accessing the data source.
Password	Password for accessing the data source.

5. Access the data source.

   When creating a data source, associate the data source with the created datasource authentication to access the data source.

   Table 10-12 lists the fields used to associate with the datasource authentication during table creation.

   Table 10-12 Fields that are used to associate with password datasource authentication during table creation

Job Type	Parameter	Mandatory	Data Type	Description
Spark SQL	passwd auth	No	String	Name of datasource authentication. It is applicable to GaussDB(DWS), RDS, DDS, and Redis data sources.
Flink OpenSource SQL	pwd_auth_name	No	String	This field is used to associate datasource authentications when source, result, and dimension tables are created. Set **pwd_auth_name** to the name of the password datasource authentication. If this parameter is set, you do not need to configure a username and a password of the data source in SQL jobs.

   For details about how to create a table, see *Data Lake Insight Syntax Reference*.

10.6 Datasource Authentication Permission Management

Scenario

Grant permissions on a datasource authentication to users so multiple user jobs can use the datasource authentication without affecting each other.

Notes

- The administrator and the datasource authentication owner have all permissions. You do not need to set permissions for them, and their datasource authentication permissions cannot be modified by other users.
- When setting datasource authentication permissions for a new user, ensure that the user group to which the user belongs has the **Tenant Guest** permission.

Granting Permissions on Datasource Connections

1. Log in to the DLI management console.
2. Choose **Datasource Connections**. On the page displayed, click **Datasource Authentication**.
3. Locate the row containing the datasource authentication to be authorized and click **Manage Permission** in the **Operation** column. The **User Permissions** page is displayed.
4. Click **Grant Permission** in the upper right corner of the page. On the **Grant Permission** dialog box displayed, grant permissions on this datasource authentication to other users.

Table 10-13 Permission granting parameters

Parameter	Description
Username	Name of the IAM user to whom permissions on the datasource connection are to be granted. **NOTE** The username is the name of an existing IAM user.
Select the permissions to be granted to the user	• Access: This permission allows you to access the datasource connection. • Update: This permission allows you to update the datasource connection. • Delete: This permission allows you to delete the datasource connection. • Grant Permission: This permission allows you to grant the datasource connection permission to other users. • Grant Permission: This permission allows you to revoke the datasource connection permission to other users. However, you cannot revoke the permissions of the datasource connection owner. • View Other User's Permissions: This permission allows you to view the datasource connection permissions of other users.

Modifying the Permissions of Current User

1. Log in to the DLI management console.
2. Choose **Datasource Connections**. On the page displayed, click **Datasource Authentication**.
3. Locate the row containing the datasource authentication to be authorized and click **Manage Permission** in the **Operation** column. The **User Permissions** page is displayed.
4. Click **Set Permission** in the **Operation** column to modify the permissions of the current user. **Table 10-13** lists the detailed permission descriptions.

NOTE

- If all options under **Set Permission** are gray, you are not allowed to change permissions on this datasource connection. You can apply to the administrator, group owner, or other users who have the permission to grant permissions for the permissions to grant and revoke the datasource authentication permissions.
- To revoke all permissions of the current user, click **Revoke Permission** in the **Operation** column. The IAM user will no longer have any permission on the datasource authentication.

11 Global Configuration

11.1 Global Variables

What Is a Global Variable?

DLI allows you to set variables that are frequently used during job development as global variables on the DLI management console. This avoids repeated definitions during job editing and reduces development and maintenance costs. Global variables can be used to replace long and difficult variables, simplifying complex parameters and improving the readability of SQL statements.

This section describes how to create a global variable.

Creating Variables

1. In the navigation pane of the DLI console, choose **Global Configuration** > **Global Variables**.
2. On the **Global Variables** page, click **Create** in the upper right corner to create a global variable.

Table 11-1 Parameters description

Parameter	Description
Variable	Name of the created global variable.
Value	Global variable value.

NOTE

- Only whitelisted users are allowed to create sensitive variables. To use this function, submit a service ticket to the administrator.
- If passwords or other sensitive information is involved, you can set variables as sensitive ones.

3. After creating a global variable, use **{{xxxx}}** in the SQL statement to replace the parameter value set as the global variable. **xxxx** indicates the variable name. For example, if you set global variable **abc** to represent the table name, replace the actual table name with **{{abc}}** in the table creation statement.

```
create table {{table_name}} (String1 String, int4 int, varchar1 varchar(10))
 partitioned by (int1 int,int2 int,int3 int)
```

NOTE

- Existing sensitive variables can only be used by their respective creators. Other common global variables are shared by users under the same account and project.
- Do not use global variables in **OPTIONS** of the table creation statements.

Modifying Variables

On the **Global Variables** page, click **Modify** in the **Operation** column of a variable to modify the variable value.

NOTE

If there are multiple global variables with the same name in the same project under an account, delete the redundant global variables to ensure that the global variables are unique in the same project. In this case, all users who have the permission to modify the global variables can change the variable values.

Deleting Variables

On the **Global Variables** page, click **Delete** in the **Operation** column of a variable to delete the variable value.

NOTE

- If there are multiple global variables with the same name in the same project under an account, delete the global variables created by the user first. If there are only unique global variables, all users who have the delete permission can delete the global variables.
- After a variable is deleted, the variable cannot be used in SQL statements.

11.2 Permission Management for Global Variables

Scenario

- You can grant permissions on a global variable to users.
- The administrator and the global variable owner have all permissions. You do not need to set permissions for them, and their global variable permissions cannot be modified by other users.
- When setting global variables for a new user, ensure that the user group the user belongs to has the **Tenant Guest** permission. For details about the Tenant Guest permission and how to apply for the permission, see .

Granting Permissions on a Global Variable to a User

1. Choose **Global Configuration** > **Global Variables**, locate the row containing the desired global variable, and click **Set Permission** in the **Operation**

column. On the **User Permissions** page displayed, you can grant permissions for the global variable, set and revoke user permissions.

2. Click **Grant Permission** in the upper right corner of the page.

Table 11-2 Global variable parameters

Parameter	Description
Username	Name of the IAM user to whom permissions on the datasource connection are to be granted. **NOTE** The username is the name of an existing IAM user.
Permissions to be granted to the user	• **Update**: This permission allows you to update the global variable. • **Delete**: This permission allows you to delete the global variable. • **Grant Permission**: This permission allows you to grant permissions of the global variable to other users. • **Revoke Permission**: This permission allows you to revoke the global variable permissions that other users have but cannot revoke the global variable owner's permissions. • **View Other User's Permissions**: This permission allows you to view the global variable permissions of other users.

Granting Global Variable Permissions

Click **Set Permission** in the **Operation** column of the sub-user to modify their permissions. **Table 11-2** lists the permission parameters.

If all permission options are grayed out, you are not allowed to change permissions on this global variable. You can apply to the administrator, group owner, or other authorized users for required permissions on the global variable.

Revoking Global Variable Permissions

Click **Revoke Permission** in the **Operation** column of a sub-user to revoke their permissions. After this operation, the sub-user does not have any permission on the global variable.

11.3 Service Authorization

Prerequisites

Only the tenant account or a subaccount of user group **admin** can authorize access.

Procedure

After entering the DLI management console, you are advised to set agency permissions to ensure that DLI can be used properly.

If you need to adjust the agency permissions, modify them on the **Service Authorization** page. For details about the required agency permissions, see **Table 11-3**.

1. Select required agency permissions and click **Update Authorization**. Only the tenant account or a subaccount of user group **admin** can authorize access. If the message "Agency permissions updated" is displayed, the update is successful.
2. Once service authorization has succeeded, an agency named **dli_admin_agency** on IAM will be created. Go to the agency list to view the details. Do not delete **dli_admin_agency**.

Table 11-3 DLI agency permissions

Permission	Details	Remarks
Tenant Administrator (global service)	**Tenant Administrator** permissions are required to access data from OBS to execute Flink jobs on DLI, for example, obtaining OBS/DWS data sources, log dump (including bucket authorization), checkpointing enabling, and job import and export.	Due to cloud service cache differences, permission setting operations require about 60 minutes to take effect.
DIS Administrator	**DIS Administrator** permissions are required to use DIS data as the data source of DLI Flink jobs.	Due to cloud service cache differences, permission setting operations require about 30 minutes to take effect.
VPC Administrator	**VPC Administrator** permissions are required to use the VPC, subnet, route, VPC peering connection, and port for DLI datasource connections.	Due to cloud service cache differences, permission setting operations require about 3 minutes to take effect.
SMN Administrator	To receive notifications when a DLI job fails, **SMN Administrator** permissions are required.	Due to cloud service cache differences, permission setting operations require about 3 minutes to take effect.

12 Permissions Management

12.1 Overview

DLI has a comprehensive permission control mechanism and supports fine-grained authentication through Identity and Access Management (IAM). You can create policies in IAM to manage DLI permissions. You can use both the DLI's permission control mechanism and the IAM service for permission management.

Application Scenarios of IAM Authentication

When using DLI on the cloud, enterprise users need to manage DLI resources (queues) used by employees in different departments, including creating, deleting, using, and isolating resources. In addition, data of different departments needs to be managed, including data isolation and sharing.

DLI uses IAM for refined enterprise-level multi-tenant management. IAM provides identity authentication, permissions management, and access control, helping you securely access to your cloud resources.

With IAM, you can use your cloud account to create IAM users for your employees, and assign permissions to the users to control their access to specific resource types. For example, some software developers in your enterprise need to use DLI resources but must not delete them or perform any high-risk operations. To achieve this result, you can create IAM users for the software developers and grant them only the permissions required for using DLI resources.

NOTE

For a new user, you need to log in for the system to record the metadata before using DLI.

IAM is free of charge. You pay only for the resources you use.

If your cloud account does not need individual IAM users for permissions management, skip this chapter.

DLI System Permissions

Table 12-1 lists all the system-defined roles and policies supported by DLI.

Type: There are roles and policies.

- Roles: A type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. Only a limited number of service-level roles are available. When using roles to grant permissions, you also need to assign other roles on which the permissions depend. However, roles are not an ideal choice for fine-grained authorization and secure access control.
- Policies: A type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization, meeting requirements for secure access control. For example, you can grant DLI users only the permissions for managing a certain type of ECSs.

Table 12-1 DLI system permissions

Role/Policy Name	Description	Category
DLI FullAccess	Full permissions for DLI.	System-defined policy
DLI ReadOnlyAccess	Read-only permissions for DLI. With read-only permissions, you can use DLI resources and perform operations that do not require fine-grained permissions. For example, create global variables, create packages and package groups, submit jobs to the default queue, create tables in the default database, create datasource connections, and delete datasource connections.	System-defined policy
Tenant Administrator	Tenant administrator • Administer permissions for managing and accessing all cloud services. After a database or a queue is created, the user can use the ACL to assign rights to other users. • Scope: project-level service	System-defined role
DLI Service Admin	DLI administrator • Administer permissions for managing and accessing the queues and data of DLI. After a database or a queue is created, the user can use the ACL to assign rights to other users. • Scope: project-level service	System-defined role

DLI Permission Types

Table 12-2 lists the DLI service permissions. For details about the resources that can be controlled by DLI, see Table 12-7.

Table 12-2 DLI permission types

Permission Type	Subtype	Console Operations	SQL Syntax	API Definition
Queue Permissions	Queue management permissions	For details, see **Queue Permission Management**.	None	For details, see "Granting Users with the Queue Usage Permission" in the *Data Lake Insight API Reference*.
	Queue usage permission			
Data Permissions	Database permissions	For details, see **Managing Database Permissions** and **Managing Table Permissions**.	For details, see **SQL Syntax of Batch Jobs** > **Data Permissions Management** > **Data Permissions List** in the *Data Lake Insight SQL Syntax Reference*.	For details, see **Permission-related APIs** > **Granting Users with the Data Usage Permission** in the *Data Lake Insight API Reference*.
	Table permissions			
	Column permissions			
Job Permissions	Flink job permissions	For details, see **Managing Flink Job Permissions**.	None	For details, see **Permission-related APIs** > **Granting Users with the Data Usage Permission** in the *Data Lake Insight API Reference*.
Package Permissions	Package group permissions	For details, see **Managing Permissions on Packages and Package Groups**.	None	For details, see **Permission-related APIs** > **Granting Users with the Data Usage Permission** in the *Data Lake Insight API Reference*.
	Package permissions			

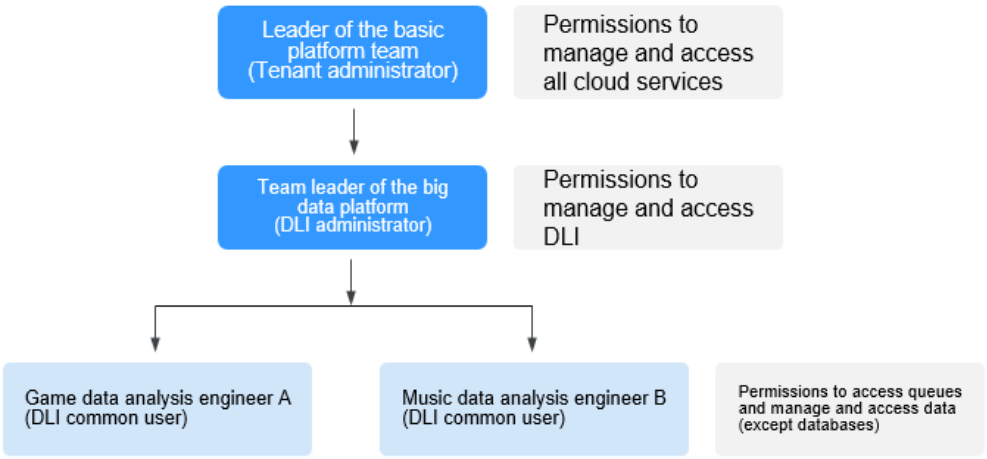
Permission Type	Subtype	Console Operations	SQL Syntax	API Definition
Datasource Connection Permissions	Datasource connection permissions	For details, see **Datasource Authentication Permission Management**.	None	For details, see **Permission-related APIs** > **Granting Users with the Data Usage Permission** in the *Data Lake Insight API Reference*.

Examples

An Internet company mainly provides game and music services. DLI is used to analyze user behaviors and assist decision making.

As shown in **Figure 12-1**, the **Leader of the Basic Platform Team** has applied for a **Tenant Administrator** account to manage and use cloud services. Since the **Big Data Platform Team** needs DLI for data analysis, the **Leader of the Basic Platform Team** adds a subaccount with the permission of **DLI Service Admin** to manage and use DLI. The **Leader of the Basic Platform Team** creates a **Queue A** and assigns it to **Data Engineer A** to analyze the gaming data. A **Queue B** is also assigned to **Data Engineer B** to analyze the music data. Besides granting the queue usage permission, the **Leader of the Basic Platform Team** grants data (except the database) management and usage permissions to the two engineers.

Figure 12-1 Granting permissions



The **Data Engineer A** creates a table named **gameTable** for storing game prop data and a table named **userTable** for storing game user data. The music service is a new service. To explore potential music users among existing game users, the **Data Engineer A** assigns the query permission on the **userTable** to the **Data**

Engineer B. In addition, **Data Engineer B** creates a table named **musicTable** for storing music copyrights information.

Table 12-3 describes the queue and data permissions of **Data Engineer A** and **Data Engineer B**.

Table 12-3 Permission description

User	Data Engineer A (game data analysis)	Data Engineer B (music data analysis)
Queues	Queue A (queue usage permission)	Queue B (queue usage permission)
Data (Table)	gameTable (table management and usage permission)	musicTable (table management and usage permission)
	userTable (table management and usage permission)	userTable (table query permission)

NOTE

The queue usage permission includes job submitting and terminating permissions.

12.2 Creating an IAM User and Granting Permissions

You can use Identity and Access Management (IAM) to implement fine-grained permissions control on DLI resources. For details, see **Overview**.

If your cloud account does not need individual IAM users, then you may skip over this chapter.

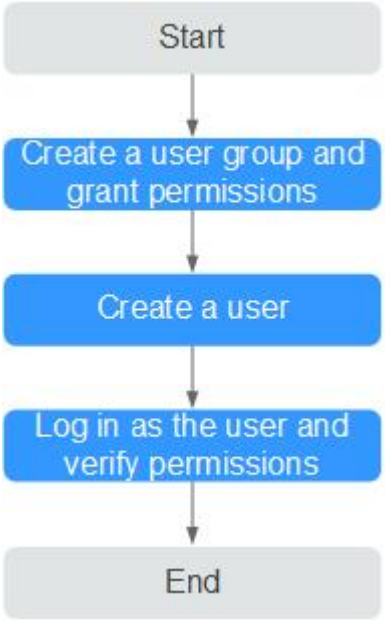
This section describes how to create an IAM user and grant DLI permissions to the user. **Figure 12-2** shows the procedure.

Prerequisites

Before assigning permissions to user groups, you should learn about system policies and select the policies based on service requirements. For details about system permissions supported by DLI, see **DLI System Permissions**.

Process Flow

Figure 12-2 Process for granting DLI permissions



1. Create a user group and grant the permission to it.

   Create a user group on the IAM console, and assign the **DLI ReadOnlyAccess** permission to the group.

2. Create a user and add the user to the user group.

   Create a user on the IAM console and add the user to the group created in **1**.

3. Log in and verify the permission.

   Log in to the management console using the newly created user, and verify that the user's permissions.

   - Choose **Service List** > **Data Lake Insight**. The DLI management console is displayed. If you can view the queue list on the **Queue Management** page but cannot buy DLI queues by clicking **Buy Queue** in the upper right corner (assume that the current permission contains only **DLI ReadOnlyAccess**), the **DLI ReadOnlyAccess** permission has taken effect.
   - Choose any other service in **Service List**. If a message appears indicating that you have insufficient permissions to access the service, the **DLI ReadOnlyAccess** permission has already taken effect.

12.3 Creating a Custom Policy

Custom policies can be created as a supplement to the system policies of DLI. You can add actions to custom policies. For the actions supported for custom policies, see "Permissions Policies and Supported Actions" in the *Elastic Volume Service API Reference*.

You can create custom policies in either of the following two ways:

- Visual editor: Select cloud services, actions, resources, and request conditions without the need to know policy syntax.
- JSON: Create a policy in the JSON format from scratch or based on an existing policy.

. This section describes common DLI custom policies.

Policy Field Description

The following example assumes that the authorized user has the permission to create tables in all databases in all regions:

```
{
   "Version": "1.1",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "dli:database:createTable"
         ],
         "Resource": [
            "dli:*:*:database:*"
         ]
      }
   ]
}
```

- Version

  1.1 indicates a fine-grained permission policy that defines permissions required to perform operations on specific cloud resources under certain conditions.

- Effect

  The value can be **Allow** and **Deny**. If both **Allow** and **Deny** are found in statements, the **Deny** overrides the **Allow**.

- Action

  Specific operation on a resource. A maximum of 100 actions are allowed.

  NOTE

  - The format is *Service name*:*Resource type*:*Action*, for example, **dli:queue:submit_job**.
  - *Service name*: product name, such as **dli**, **evs**, and **vpc**. Only lowercase letters are allowed. Resource types and operations are not case-sensitive. You can use an asterisk (*) to represent all operations.
  - *Resource type*: For details, see **Table 12-7**.
  - *Action*: action registered in IAM.

- Condition

  Conditions determine when a policy takes effect. A condition consists of a condition key and operator.

  A condition key is a key in the **Condition** element of a statement. There are global and service-level condition keys.

  – Global condition keys (prefixed with **g:**) apply to all actions. For details, see condition key description in Policy Syntax.

  – Service-level condition keys apply only to operations of the specific service.

An operator is used together with a condition key to form a complete condition statement. For details, see **Table 12-4**.

IAM provides a set of DLI predefined condition keys. The following table lists the predefined condition keys of DLI.

Table 12-4 DLI request conditions

Condition Key	Type	Operator	Description
g:CurrentTime	Global	Date and time	Time when an authentication request is received **NOTE** The time is expressed in the format defined by **ISO 8601**, for example, **2012-11-11T23:59:59Z**.
g:MFAPresent	Global	Boolean	Whether multi-factor authentication is used during user login
g:UserId	Global	String	ID of the current login user
g:UserName	Global	String	Current login user
g:ProjectName	Global	String	Project that you have logged in to
g:DomainName	Global	String	Domain that you have logged in to

- Resource

  The format is *Service name*:*Region*:*Domain ID*:*Resource type*:*Resource path*. The wildcard (*) indicates all options. For details about the resource types and path, see **Table 12-7**.

  Example:

  dli:*:*:queue:* indicates all queues.

Creating a Custom Policy

You can set actions and resources of different levels based on scenarios.

1. Define an action.

   The format is *Service name*:*Resource type*:*Action*. The wildcard is *. Example:

   Table 12-5 Action

Action	Description
dli:queue:submit_job	Submission operations on a DLI queue

Action	Description
dli:queue:*	All operations on a DLI queue
dli:*:*	All operations on all DLI resource types

For more information about the relationship between operations and system permissions, see **Common Operations Supported by DLI System Policy**.

2. Define a resource.

   The format is *Service name*:*Region*:*Domain ID*:**Resource type:Resource path**. The wildcard (*) indicates all resources. The five fields can be flexibly set. Different levels of permission control can be set for resource paths based on scenario requirements. If you need to set all resources of the service, you do not need to specify this field. For details about the definition of Resource, see **Table 12-6**. For details about the resource types and paths in Resource, see **Table 12-7**.

Table 12-6 Resource

Resource	Description
DLI:*:*:table:databases.dbname.tables.*	DLI, any region, any account ID, all table resources of database **dbname**
DLI:*:*:database:databases.dbname	DLI, any region, any account ID, resource of database **dbname**
DLI:*:*:queue:queues.*	DLI, any region, any account ID, any queue resource
DLI:*:*:jobs:jobs.flink.1	DLI, any region, any account ID, Flink job whose ID is 1

Table 12-7 DLI resources and their paths

Resource Type	Resource Names	Path
queue	DLI queue	queues.queuename
database	DLI database	databases.dbname
table	DLI table	databases.dbname.tables.tbname
column	DLI column	databases.dbname.tables.tbname.columns.colname
jobs	DLI Flink job	jobs.flink.jobid
resource	DLI package	resources.resourcename

Resource Type	Resource Names	Path
group	DLI package group	groups.groupname
datasource auth	DLI cross-source authentication information	datasourceauth.name
edsconnections	Enhanced datasource connection	edsconnections.*connection ID*

3. Combine all the preceding fields into a JSON file to form a complete policy. You can set multiple actions and resources. You can also create a policy on the visualized page provided by IAM. For example:

   The authorized user has the permission to create and delete any database, submit jobs for any queue, and delete any table under any account ID in any region of DLI.

```
{
   "Version": "1.1",
   "Statement": [
      {
         "Effect": " Allow",
         "Action": [
               "dli:database:createDatabase",
               "dli:database:dropDatabase",
               "dli:queue:submitJob",
               "dli:table:dropTable"
         ],
         "Resource": [
               "dli:*:*:database:*",
               "dli:*:*:queue:*",
               "dli:*:*:table:*"
         ]
      }
   ]
}
```

Example Custom Policies

- Example 1: Allow policies
  - Allow users to create tables in all databases of all regions:

```
{
   "Version": "1.1",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "dli:database:createTable"
         ],
         "Resource": [
            "dli:*:*:database:*"
         ]
      }
   ]
}
```

  - Allow users to query column **col** in the table **tb** of the database **db**:

```
{
   "Version": "1.1",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "dli:column:select"
         ],
         "Resource": [
            "dli:*:*:column:databases.db.tables.tb.columns.col"
         ]
      }
   ]
}
```

- Example 2: Deny policies

  A deny policy must be used together with other policies. That is, a user can set a deny policy only after being assigned some operation permissions. Otherwise, the deny policy does not take effect.

  If the permissions assigned to a user contain both Allow and Deny actions, the Deny actions take precedence over the Allow actions.

  - Deny users to create or delete databases, submit jobs (except the default queue), or delete tables.

```
{
   "Version": "1.1",
   "Statement": [
      {
         "Effect": "Deny",
         "Action": [
            "dli:database:createDatabase",
            "dli:database:dropDatabase",
            "dli:queue:submitJob",
            "dli:table:dropTable"
         ],
         "Resource": [
            "dli:*:*:database:*",
            "dli:*:*:queue:*",
            "dli:*:*:table:*"
         ]
      }
   ]
}
```

  - Deny users to submit jobs in the demo queue.

```
{
   "Version": "1.1",
   "Statement": [
      {
         "Effect": "Deny",
         "Action": [
            "dli:queue:submitJob"
         ],
         "Resource": [
            "dli:*:*:queue:queues.demo"
         ]
      }
   ]
}
```

12.4 DLI Resources

A resource is an object that exists within a service. You can select DLI resources by specifying their paths.

Table 12-8 DLI resources and their paths

Resource Type	Resource Names	Path
queue	DLI queue	queues.queuename
database	DLI database	databases.dbname
table	DLI table	databases.dbname.tables.tbname
column	DLI column	databases.dbname.tables.tbname.columns.colname
jobs	DLI Flink job	jobs.flink.jobid
resource	DLI package	resources.resourcename
group	DLI package group	groups.groupname
datasourceauth	DLI cross-source authentication information	datasourceauth.name
edsconnections	Enhanced datasource connection	edsconnections.*connection ID*

12.5 DLI Request Conditions

Request conditions are useful in determining when a custom policy takes effect. A request condition consists of a condition key and operator. Condition keys are either global or service-level and are used in the Condition element of a policy statement. Global condition keys (starting with **g:**) are available for operations of all services, while service-level condition keys (starting with a service name such as **dli**) are available only for operations of a specific service. An operator is used together with a condition key to form a complete condition statement.

IAM provides a set of DLI predefined condition keys. The following table lists the predefined condition keys of *DLI*.

Table 12-9 DLI request conditions

Condition Key	Type	Operator	Description
g:CurrentTime	Global	Date and time	Time when an authentication request is received **NOTE** The time is expressed in the format defined by **ISO 8601**, for example, **2012-11-11T23:59:59Z**.

Condition Key	Type	Operator	Description
g:MFAPresent	Global	Boolean	Whether multi-factor authentication is used during user login
g:UserId	Global	String	ID of the current login user
g:UserName	Global	String	Current login user
g:ProjectName	Global	String	Project that you have logged in to
g:DomainName	Global	String	Domain that you have logged in to

12.6 Common Operations Supported by DLI System Policy

Table 12-10 lists the common operations supported by each system policy of DLI. Choose proper system policies according to this table. For details about the SQL statement permission matrix in DLI in terms of permissions on databases, tables, and roles, see **SQL Syntax of Batch Jobs** > **Data Permissions Management** > **Data Permissions List** in the *Data Lake Insight SQL Syntax Reference*.

Table 12-10 Common operations supported by each system policy

Resources	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
Queue	DROP_QUEUE	Deleting a queue	√	×	√	√
	SUBMIT_JOB	Submitting the job	√	×	√	√
	CANCEL_JOB	Terminating the job	√	×	√	√
	RESTART	Restarting a queue	√	×	√	√
	GRANT_PRIVILEGE	Granting permissions to the queue	√	×	√	√

Resources	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
	REVOKE_PRIVILEGE	Revoking permissions from the queue	√	×	√	√
	SHOW_PRIVILEGES	Viewing the queue permissions of other users	√	×	√	√
Database	DROP_DATABASE	Deleting a database	√	×	√	√
	CREATE_TABLE	Creating a table	√	×	√	√
	CREATE_VIEW	Creating a view	√	×	√	√
	EXPLAIN	Explaining the SQL statement as an execution plan	√	×	√	√
	CREATE_ROLE	Creating a role	√	×	√	√
	DROP_ROLE	Deleting a role	√	×	√	√
	SHOW_ROLES	Displaying a role	√	×	√	√
	GRANT_ROLE	Binding a role	√	×	√	√
	REVOKE_ROLE	Unbinding the role	√	×	√	√
	SHOW_USERS	Displaying the binding relationships between all roles and users	√	×	√	√
	GRANT_PRIVILEGE	Granting permissions to the database	√	×	√	√

Resources	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
	REVOKE_PRIVILEGE	Revoking permissions to the database	√	×	√	√
	SHOW_PRIVILEGES	Viewing database permissions of other users	√	×	√	√
	DISPLAY_ALL_TABLES	Displaying tables in the database	√	√	√	√
	DISPLAY_DATABASE	Displaying databases	√	√	√	√
	CREATE_FUNCTION	Creating a function	√	×	√	√
	DROP_FUNCTION	Deleting a function	√	×	√	√
	SHOW_FUNCTIONS	Displaying all functions	√	×	√	√
	DESCRIBE_FUNCTION	Displaying function details	√	×	√	√
Table	DROP_TABLE	Deleting a table	√	×	√	√
	SELECT	Querying a table	√	×	√	√
	INSERT_INTO_TABLE	Inserting	√	×	√	√
	ALTER_TABLE_ADD_COLUMNS	Adding a column	√	×	√	√
	INSERT_OVERWRITE_TABLE	Rewriting	√	×	√	√
	ALTER_TABLE_RENAME	Renaming a table	√	×	√	√

Resources	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
	ALTER_TABLE_ADD_PARTITION	Adding partitions to the partition table	√	×	√	√
	ALTER_TABLE_RENAME_PARTITION	Renaming a table partition	√	×	√	√
	ALTER_TABLE_DROP_PARTITION	Deleting partitions from a partition table	√	×	√	√
	SHOW_PARTITIONS	Displaying all partitions	√	×	√	√
	ALTER_TABLE_RECOVER_PARTITION	Restoring table partitions	√	×	√	√
	ALTER_TABLE_SET_LOCATION	Setting the partition path	√	×	√	√
	GRANT_PRIVILEGE	Granting permissions to the table	√	×	√	√
	REVOKE_PRIVILEGE	Revoking permissions from the table	√	×	√	√
	SHOW_PRIVILEGES	Viewing table permissions of other users	√	×	√	√
	DISPLAY_TABLE	Displaying a table	√	√	√	√
	DESCRIBE_TABLE	Displaying table information	√	×	√	√

13 Other Common Operations

13.1 Importing Data to a DLI Table

Importing Data Using OBS

On the DLI management console, you can import data stored on OBS to DLI tables from **Data Management > Databases and Tables > Table Management** and **SQL Editor** pages. For details, see **Importing Data to the Table**.

Importing Data Using CDM

Use the Cloud Data Migration (CDM) service to import data from OBS to DLI. You need to create a CDM queue first.

For details about how to create the queue, see "Migrating Data from OBS to DLI" in the *Cloud Data Migration User Guide*.

Pay attention to the following configurations:

- The VPC to which the DLI account belongs is the same as the VPC of the CDM queue.
- You need to create two links, including a DLI link and an OBS link.
- The format of the file to be transmitted can be **CSV** or **JSON**.

13.2 Viewing Monitoring Metrics

Description

This section describes metrics reported by DLI to Cloud Eye as well as their namespaces and dimensions. You can use the management console or APIs provided by Cloud Eye to query the metrics of the monitored object and alarms generated for DLI.

Namespace

SYS.DLI

Metric

Table 13-1 DLI metrics

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
queue_cu_num	CU usage of a queue	Displays the number of CUs applied by the user queue	≥ 0	Queues	5 minutes
queue_job_launching_num	Number of Jobs Being Submitted	Displays the number of jobs in the Submitting state in the user queue.	≥ 0	Queues	5 minutes
queue_job_running_num	Number of Running Jobs	Displays the number of running jobs in the user queue.	≥ 0	Queues	5 minutes
queue_job_succeed_num	Number of Finished Jobs	Displays the number of completed jobs in the user queue.	≥ 0	Queues	5 minutes
queue_job_failed_num	Failed Jobs	Displays the number of failed jobs in the user queue.	≥ 0	Queues	5 minutes

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
queue_job_cancelled_num	Number of Canceled Jobs	Displays the number of canceled jobs in the user queue.	≥ 0	Queues	5 minutes
queue_cpu_usage	Queue CPU Usage	Displays the CPU usage of user queues.	0–100	Queues	5 minutes
queue_disk_usage	Queue Disk Usage	Displays the disk usage of user queues.	0–100	Queues	5 minutes
queue_disk_used	Max Disk Usage	Displays the maximum disk usage of user queues.	0~100	Queues	5 minutes
queue_mem_usage	Queue Memory Usage	Displays the memory usage of user queues.	0–100	Queues	5 minutes
queue_mem_used	Used Memory	Displays the memory usage rate of the user queues.	≥ 0	Queues	5 minutes
flink_read_records_per_second	Flink Job Data Read Rate	Displays the data input rate of a Flink job for monitoring and debugging.	≥ 0	Flink jobs	10 seconds

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
flink_write_records_per_second	Flink Job Data Write Rate	Displays the data output rate of a Flink job for monitoring and debugging.	≥ 0	Flink jobs	10 seconds
flink_read_records_total	Flink Job Total Data Read	Displays the total number of data inputs of a Flink job for monitoring and debugging.	≥ 0	Flink jobs	10 seconds
flink_write_records_total	Flink Job Total Data Write	Displays the total number of output data records of a Flink job for monitoring and debugging.	≥ 0	Flink jobs	10 seconds
flink_read_bytes_per_second	Flink Job Byte Read Rate	Displays the number of input bytes per second of a Flink job.	≥ 0	Flink jobs	10 seconds
flink_write_bytes_per_second	Flink Job Byte Write Rate	Displays the number of output bytes per second of a Flink job.	≥ 0	Flink jobs	10 seconds

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
flink_read_bytes_total	Flink Job Total Read Byte	Displays the total number of input bytes of a Flink job.	≥ 0	Flink jobs	10 seconds
flink_write_bytes_total	Flink Job Total Write Byte	Displays the total number of output bytes of a Flink job.	≥ 0	Flink jobs	10 seconds
flink_cpu_usage	Flink Job CPU Usage	Displays the CPU usage of Flink jobs.	0–100	Flink jobs	10 seconds
flink_mem_usage	Flink Job Memory Usage	Displays the memory usage of Flink jobs.	0–100	Flink jobs	10 seconds
flink_max_op_latency	Flink Job Max Operator Latency	Displays the maximum operator delay of a Flink job. The unit is **ms**.	≥ 0	Flink jobs	10 seconds

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
flink_max_op_backpressure_level	Flink Job Maximum Operator Backpressure	Displays the maximum operator backpressure value of a Flink job. A larger value indicates severer backpressure. **0**: OK **50**: low **100**: high	0–100	Flink jobs	10 seconds

Dimension

Table 13-2 Dimension

Key	Value
queue_id	Queue
flink_job_id	Flink job

Viewing DLI Monitoring Metrics on Cloud Eye

1. Search for Cloud Eye on the management console.
2. In the navigation pane on the left of the Cloud Eye console, click **Cloud Service Monitoring** > **Data Lake Insight**.
3. Select a queue to view its information.

13.3 DLI Operations That Can Be Recorded by CTS

With CTS, you can record operations associated with DLI for later query, audit, and backtrack operations.

Table 13-3 DLI operations that can be recorded by CTS

Operation	Resource Type	Trace Name
Creating a database	database	createDatabase
Deleting a database	database	deleteDatabase
Modifying the Database Owner	database	alterDatabaseOwner
Creating a table	table	createTable
Deleting tables	table	deleteTable
Exporting table data	table	exportData
Importing table data	table	importData
Modifying the owner of a table	table	alterTableOwner
Creating a queue	queue	createQueue
Deleting a queue	queue	dropQueue
Granting permissions to a queue	queue	shareQueue
Modifying a Queue CIDR Block	queue	replaceQueue
Restarting a queue	queue	queueActions
Scaling out/in a queue	queue	queueActions
Submitting a job	queue	submitJob
Canceling a job	queue	cancelJob
Granting DLI the permission to access OBS buckets	obs	obsAuthorize
Checking the SQL syntax	job	checkSQL
Creating a job	job	createJob
Updating a job	job	updateJob
Deleting a job	job	deleteJob
Purchasing CUH packages	order	orderPackage
Freezing resources	resource	freezeResource
Unfreezing resources	resource	unfreezeResource
Terminating resources	resource	deleteResource
Clearing resources	resource	cleanResource

Operation	Resource Type	Trace Name
Granting data permissions	data	dataAuthorize
Granting permissions on other projects	data	authorizeProjectData
Exporting query results	data	storeJobResult
Saving a SQL template	sqlTemplate	saveSQLTemplate
Updating a SQL template	sqlTemplate	updateSQLTemplate
Deleting a SQL template	sqlTemplate	deleteSQLTemplate
Creating a Flink template	flinkTemplate	createStreamTemplate
Updating a Flink template	flinkTemplate	createStreamTemplate
Deleting a Flink template	flinkTemplate	deleteStreamTemplate
Creating a data upload task	uploader	createUploadJob
Obtaining the authentication to perform a data upload task	uploader	getUploadAuthInfo
Submitting a data upload task	uploader	commitUploadJob
Creating a datasource authentication and uploading a certificate	authInfo	uploadAuthInfo
Updating a datasource authentication	authInfo	updateAuthInfop
Deleting a datasource authentication	authInfo	deleteAuthInfo
Updating the quota	quota	updateQuota
Uploading a resource package	pkgResource	uploadResources
Deleting a resource package	pkgResource	deleteResource
Creating a basic datasource connection	datasource	createDatasourceConn
Deleting a basic datasource connection	datasource	deleteDatasourceConn
Reactivating a basic datasource connection	datasource	reactivateDSConnection

Operation	Resource Type	Trace Name
Creating an enhanced datasource connection	datasource	createConnection
Deleting an enhanced datasource connection	datasource	getConnection
Binding a queue	datasource	associateQueueToDatasourceConn
Unbinding a queue	datasource	disassociateQueueToDatasourceConn
Modifying the host information	datasource	updateHostInfo
Adding a route	datasource	addRoute
Deleting a route	datasource	deleteRoute
Creating a topic	smn	createTopic
Creating an agency	agency	createAgencyV2
Creating a batch processing job	batch	createBatch
Canceling a batch processing job	batch	cancelBatch
Creating a session	session	createSession
Deleting a session	session	deleteSession
Creating a statement	statement	createStatement
Canceling execution of a statement	statement	cancelStatement
Creating a global variable	globalVar	createGlobalVariable
Deleting a global variable	globalVar	deleteGlobalVariable
Modifying a global variable	globalVar	updateGlobalVariable

13.4 Quotas

What Is a Quota?

A quota limits the quantity of a resource available to users, thereby preventing spikes in the usage of the resource.

You can also request for an increased quota if your existing quota cannot meet your service requirements.

How Do I View My Quotas?

1. Log in to the management console.
2. Click  in the upper left corner and select **Region** and **Project**.
3. Click  (the **My Quotas** icon) in the upper right corner.
   The **Service Quota** page is displayed.
4. View the used and total quota of each type of resources on the displayed page.
   If a quota cannot meet service requirements, increase a quota.

How Do I Apply for a Higher Quota?

The system does not support online quota adjustment. To increase a resource quota, dial the hotline or send an email to the customer service. We will process your application and inform you of the progress by phone call or email.

Before you contact customer service, prepare the following information:

- Account name, project name, and project ID
  Log in to the management console, click the username in the upper-right corner, choose **My Credentials**, and obtain the domain name, project name, and project ID.
- Quota information, including:
  - Service name
  - Quota type
  - Required quota

Learn how to obtain the service hotline and email address.

14 FAQ

14.1 Flink Jobs

14.1.1 What Data Formats and Data Sources Are Supported by DLI Flink Jobs?

DLI Flink jobs support the following data formats:

Avro, Avro_merge, BLOB, CSV, EMAIL, JSON, ORC, Parquet, and XML.

DLI Flink jobs support data from the following data sources:

CloudTable HBase, CloudTable OpenTSDB, CSS Elasticsearch, DCS, DDS, DIS, DMS, GaussDB(DWS), EdgeHub, MRS HBase, MRS Kafka, open-source Kafka, file systems, OBS, RDS, and SMN

14.1.2 How Do I Authorize a Subuser to View Flink Jobs?

A sub-user can view queues but cannot view Flink jobs. You can authorize the sub-user using DLI or IAM.

- Authorization on DLI
  a. Log in to the DLI console using a tenant account, a job owner account, or an account with the **DLI Service Administrator** permission.
  b. Choose **Job Management** > **Flink Jobs**. On the displayed page, locate the target job.
  c. In the **Operation** column of the target job, choose **More** > **Permissions**.
  d. On the displayed page, click **Grant Permission**. Enter the name of the user to be authorized and select the required permissions. Click **OK**. The authorized user can view the job and perform related operations.
- Authorization on IAM
  a. Log in to the IAM console. In the navigation pane, choose **Permissions** > **Policies/Roles**. On the displayed page, click **Create Custom Policy**.
  b. Create a permission policy for the subuser to view DLI Flink jobs.

- **Policy Name**: Use the default name or customize a name.
- **Scope**: Select **Project-level services**.
- **Policy View**: Select **Visual editor**.
- **Policy Content**: Select **Allow**, **Data Lake Insight**, and **dli:jobs:list_all** in sequence.

Click **OK** to create the policy.

c. Go to the **User Group** page, locate the user group to which the subuser to be authorized belongs and click the user group name. On the displayed page, click **Assign**.

d. Grant permissions to the user group.

- Select **Region-specific projects** for **Scope**.
- Select the permission policy created in **b** for **Permissions**.

  You can also select **DLI Service Admin** (with all DLI permissions) for the subuser to view Flink jobs.

14.1.3 How Do I Set Auto Restart upon Exception for a Flink Job?

Scenario

DLI Flink jobs are highly available. You can enable the automatic restart function to automatically restart your jobs after short-time faults of peripheral services are rectified.

Procedure

1. Log in to the DLI console. In the navigation pane, choose **Job Management** > **Flink Jobs**.
2. Click a job name and select **Auto-restart upon exception** on the job editing page.

14.1.4 How Do I Save Flink Job Logs?

When you create a Flink SQL job or Flink Jar job, you can select **Save Job Log** on the job editing page to save job running logs to OBS.

To set the OBS bucket for storing the job logs, specify a bucket for **OBS Bucket**. If the selected OBS bucket is not authorized, click **Authorize**.

The logs are saved in the following path: *Bucket name*/**jobs/logs/***Directory starting with the job ID*. You can customize the bucket name in the path. **/jobs/logs/***Directory starting with the job ID* is a fixed format.

In the job list, click the job name. In the **Run Log** tab, click the provided OBS link to go to the path.

14.1.5 How Can I Check Flink Job Results?

- DLI can output Flink job results to RDS. You can view the results in RDS. For details, see *Relational Database Service Getting Started*.
- DLI can output Flink job results to SMN, and SMN sends the results to the user's terminal. For details, see *Simple Message Notification Getting Started*.
- DLI can output Flink job results to Kafka. You can view the results in Kafka clusters. For details, visit the **Kafka official website**.
- DLI can output Flink job results to CloudTable. You can view the results in CloudTable. For details, see *CloudTable Service User Guide*.
- DLI can export Flink job results to CSS. You can view the results in CSS. For details, see "Getting Started" in *Cloud Search Service User Guide*.
- DLI can export Flink job results to DCS. You can view the results in DCS. For details, see *Distributed Cache Service User Guide*.

14.1.6 Why Is Error "No such user. userName:xxxx." Reported on the Flink Job Management Page When I Grant Permission to a User?

Symptom

Choose **Job Management** > **Flink Jobs**. In the **Operation** column of the target job, choose **More** > **Permissions**. When a new user is authorized, **No such user. userName:xxxx.** is displayed.

Solution

Check whether the username exists and whether the user has logged in to the system before authorization.

14.1.7 How Do I Know Which Checkpoint the Flink Job I Stopped Will Be Restored to When I Start the Job Again?

Symptom

Checkpoint was enabled when a Flink job is created, and the OBS bucket for storing checkpoints was specified. After a Flink job is manually stopped, no message is displayed specifying the checkpoint where the Flink job will be restored if the Flink job is started again.

Solution

The generation mechanism and format of Flink checkpoints are the same as those of savepoints. You can import a savepoint of the job to restore it from the latest checkpoint saved in OBS.

1. Log in to the DLI console. In the navigation pane, choose **Job Management** > **Flink Jobs**.
2. Locate the row that contains the target Flink job, and click **Import Savepoint** in the **Operation** column.

3. In the displayed dialog box, select the OBS bucket path storing the checkpoint. The checkpoint save path is *Bucket name*/**jobs/checkpoint/***directory starting with the job ID*. Click **OK**.
4. Start the Flink job again. The job will be restored fom the imported savepoint.

14.1.8 Why Is a Message Displayed Indicating That the SMN Topic Does Not Exist When I Use the SMN Topic in DLI?

When you set running parameters of a DLI Flink job, you can enable **Alarm Generation upon Job Exception** to receive alarms when the job runs abnormally or is in arrears.

If the SMN topic you select for your DLI job does not exist, log in to the Identity and Access Management (IAM) console, select the user group to which the IAM user belongs, and add the SMN policy for your region to the group.

14.1.9 How Much Data Can Be Processed in a Day by a Flink SQL Job?

The consumption capability of a Flink SQL job depends on the data source transmission, queue size, and job parameter settings. The peak consumption is 10 Mbit/s.

14.1.10 Does Data in the Temporary Stream of Flink SQL Need to Be Cleared Periodically? How Do I Clear the Data?

The temp stream in Flink SQL is similar to a subquery. It is a logical stream used to simplify the SQL logic and does not generate data storage. Therefore, there is no need to clean data.

14.1.11 Why Is a Message Displayed Indicating That the OBS Bucket Is Not Authorized When I Select an OBS Bucket for a Flink SQL Job?

- Symptom

  When you create a Flink SQL job and configure the parameters, you select an OBS bucket you have created. The system displays a message indicating that the OBS bucket is not authorized. After you click **Authorize**, the system displays a message indicating that an internal error occurred on the server and you need to contact customer service or try again later.

- Solution

  On the settings page, press F12 to view the error details. The following is an example:

  ```
  {"error_msg":"An internal error occurred. {0} Contact customer services or try again later ","error_json_opt":{"error": "Unexpected exception[NoSuchElementException: None.get]"},"error_code":"DLI.10001"}
  ```

  Check whether a DLI agency has been created. If you do not have the permission to create an agency. On the DLI console, choose **Global Configuration** > **Service Authorization**, select **Tenant Administrator (Global service)**, and click **Update**.

14.1.12 How Do I Create an OBS Partitioned Table for a Flink SQL Job?

Scenario

When using a Flink SQL job, you need to create an OBS partition table for subsequent batch processing.

Procedure

In the following example, the **day** field is used as the partition field with the parquet encoding format (only the parquet format is supported currently) to dump **car_info** data to OBS.

```
create sink stream car_infos (
  carId string,
  carOwner string,
  average_speed double,
  day string
  ) partitioned by (day)
  with (
    type = "filesystem",
    file.path = "obs://obs-sink/car_infos",
    encode = "parquet",
    ak = "{{myAk}}",
    sk = "{{mySk}}"
);
```

Structure of the data storage directory in OBS: **obs://obs-sink/car_infos/day=xx/part-x-x**.

After the data is generated, the OBS partition table can be established for subsequent batch processing through the following SQL statements:

1. Create an OBS partitioned table.

   ```
   create table car_infos (
     carId string,
     carOwner string,
     average_speed double
   )
     partitioned by (day string)
     stored as parquet
     location 'obs://obs-sink/car-infos';
   ```

2. Restore partition information from the associated OBS path.

   ```
   alter table car_infos recover partitions;
   ```

14.1.13 How Do I Dump Data to OBS and Create an OBS Partitioned Table?

In this example, the **day** field is used as the partition field with the parquet encoding format (only the parquet format is supported currently) to dump **car_info** data to OBS. For more information, see "File System Sink Stream (Recommended)" in *Data Lake Insight SQL Syntax Reference*.

```
create sink stream car_infos (
  carId string,
  carOwner string,
  average_speed double,
  day string
  ) partitioned by (day)
```

```
 with (
   type = "filesystem",
   file.path = "obs://obs-sink/car_infos",
   encode = "parquet",
   ak = "{{myAk}}",
   sk = "{{mySk}}"
);
```

Structure of the data storage directory in OBS: **obs://obs-sink/car_infos/day=xx/part-x-x**.

After the data is generated, the OBS partition table can be established for subsequent batch processing through the following SQL statements:

1. Create an OBS partition table.
   ```
   create table car_infos (
     carId string,
     carOwner string,
     average_speed double
   )
     partitioned by (day string)
     stored as parquet
     location 'obs://obs-sink/car-infos';
   ```
2. Restore partition information from the associated OBS path.
   ```
   alter table car_infos recover partitions;
   ```

14.1.14 Why Is Error Message "DLI.0005" Displayed When I Use an EL Expression to Create a Table in a Flink SQL Job?

Symptom

When I run the creation statement with an EL expression in the table name in a Flink SQL job, the following error message is displayed:

```
DLI.0005: AnalysisException: t_user_message_input_#{date_format(date_sub(current_date(), 1), 'yyyymmddhhmmss')} is not a valid name for tables/databases. Valid names only contain alphabet characters, numbers and _.
```

Solution

Replace the number sign (#) in the table name to the dollar sign ($). The format of the EL expression used in DLI should be **${*expr*}**.

14.1.15 Why Is No Data Queried in the DLI Table Created Using the OBS File Path When Data Is Written to OBS by a Flink Job Output Stream?

Symptom

After data is written to OBS through the Flink job output stream, data cannot be queried from the DLI table created in the OBS file path.

For example, use the following Flink result table to write data to the **obs://obs-sink/car_infos** path in OBS.

```
create sink stream car_infos_sink (
  carId string,
  carOwner string,
  average_speed double,
```

```
  buyday string
  ) partitioned by (buyday)
  with (
    type = "filesystem",
    file.path = "obs://obs-sink/car_infos",
    encode = "parquet",
    ak = "{{myAk}}",
    sk = "{{mySk}}"
);
```

Use the following statements to create a DLI partition table with data retrieved from the OBS file path. No data is found when you query the **car_infos** table on DLI.

```
create table car_infos (
  carId string,
  carOwner string,
  average_speed double
)
  partitioned by (buyday string)
  stored as parquet
  location 'obs://obs-sink/car_infos';
```

Solution

1. Check whether checkpointing is enabled for the Flink result table (**car_infos_sink** in the preceding example) when you create the job on DLI. If checkpointing is disabled, enable it and run the job again to generate OBS data files.

   To enable checkpointing, perform the following steps:

   a. Log in to the DLI management console. Choose **Job Management** > **Flink Jobs** in the left navigation pane. Locate the row that contains the target Flink job and click **Edit** in the **Operation** column.

   b. In the **Running Parameters** area, check whether **Enable Checkpointing** is enabled.

2. Check whether the structure of the Flink result table is the same as that of the DLI partitioned table. For the preceding example, check whether the fields of the **car_infos_sink** table are consistent with those of the **car_infos** table.

3. Check whether the partitioning information of the the partitioned table is restored after it is created using the OBS file. The following statement restore partitions of the **car_infos** table:

   ```
   alter table car_infos recover partitions;
   ```

14.1.16 Why Does a Flink SQL Job Fails to Be Executed, and Is "connect to DIS failed java.lang.IllegalArgumentException: Access key cannot be null" Displayed in the Log?

Symptom

After a Flink SQL job is submitted on DLI, the job fails to be executed. The following error information is displayed in the job log:

```
connect to DIS failed java.lang.IllegalArgumentException: Access key cannot be null
```

Possible Causes

When configuring job running parameters for the Flink SQL job, Save Job Log or Checkpointing is enabled, and an OBS bucket for saving job logs and Checkpoints

is configured. However, the IAM user who runs the Flink SQL job does not have the OBS write permission.

Solution

1. Log in to the IAM console, search for the IAM user who runs the job in the upper left corner of the **Users** page.
2. Click the desired username to view the user group where the user belongs.
3. In the navigation pane on the left, choose **User Groups**, and search for the user group of the target user. Click the user group name, and view the permissions of the current user in **Permissions**.
4. Check whether the user group has the permission to write data to OBS, for example, **OBS OperateAccess**. If the user group does not have the OBS write permission, grant the permission to the user group.
5. Wait for 5 to 10 minutes for the permission to take effect. Run the Flink SQL job again and check the job running status.

14.1.17 Why Is Error "Not authorized" Reported When a Flink SQL Job Reads DIS Data?

Symptom

Semantic verification for a Flink SQL job (reading DIS data) fails. The following information is displayed when the job fails:

```
Get dis channel xxx info failed. error info: Not authorized, please click the overview page to do the authorize action
```

Possible Causes

Before running a Flink job, the permission to obtain DIS data is not granted to the user.

Solution

1. Log in to the DLI management console. Choose **Global Configuration** > **Service Authorization** in the navigation pane on the left.
2. On the **Service Authorization** page, select **DIS Administrator** and click **Update**.
3. Choose **Job Management** > **Flink Jobs**. On the displayed page, locate the desired Flink SQL job and restart the job.

14.1.18 Data Writing Fails After a Flink SQL Job Consumed Kafka and Sank Data to the Elasticsearch Cluster

Symptom

After a Flink SQL job consumed Kafka and sent data to the Elasticsearch cluster, the job was successfully executed, but no data is available.

Cause Analysis

Possible causes are as follows:

- The data format is incorrect.
- The data cannot be processed.

Procedure

Step 1 Check the task log on the Flink UI. The JSON body is contained in the error message, indicating that the data format is incorrect.

Step 2 Check the data format. The Kafka data contains nested JSON bodies, which cannot be parsed.

Step 3 Use either of the following methods to solve the problem:

- Create a JAR file with the UDF.
- Modify the configuration data.

Step 4 Change the data format and execute the job again.

----End

14.1.19 How Do I Configure Checkpoints for Flink Jar Jobs and Save the Checkpoints to OBS?

The procedure is as follows:

1. Add the following code to the JAR file code of the Flink Jar job:

```
// Configure the pom file on which the StreamExecutionEnvironment depends.
StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment();

        env.getCheckpointConfig().setCheckpointingMode(CheckpointingMode.EXACTLY_ONCE);
        env.getCheckpointConfig().setCheckpointInterval(40000);

env.getCheckpointConfig().enableExternalizedCheckpoints(CheckpointConfig.ExternalizedCheckpointCl
eanup.RETAIN_ON_CANCELLATION);
        RocksDBStateBackend rocksDbBackend = new RocksDBStateBackend(new
FsStateBackend("obs://${bucket}/jobs/checkpoint/my_jar"), false);
        rocksDbBackend.setOptions(new OptionsFactory() {
            @Override
            public DBOptions createDBOptions(DBOptions currentOptions) {
                return currentOptions
                        .setMaxLogFileSize(64 * 1024 * 1024)
                        .setKeepLogFileNum(3);
            }

            @Override
            public ColumnFamilyOptions createColumnOptions(ColumnFamilyOptions currentOptions) {
                return currentOptions;
            }
        });
        env.setStateBackend(rocksDbBackend);
```

NOTE

The preceding code saves the checkpoint to the **${bucket}** bucket in **jobs/checkpoint/my_jar** path every 40 seconds in **EXACTLY_ONCE** mode.

Pay attention to the checkpoint storage path. Generally, the checkpoint is stored in the OBS bucket. The path format is as follows:

- Path format: **obs://$***{bucket}*/*xxx*/*xxx*/*xxx*
- Add the following configuration to the POM file for the packages on which the StreamExecutionEnvironment depends:

```
<dependency>
        <groupId>org.apache.flink</groupId>
        <artifactId>flink-streaming-java_${scala.binary.version}</artifactId>
        <version>${flink.version}</version>
        <scope>provided</scope>
</dependency>
```

2. Configure **Runtime Configuration** and **Restore Job from Checkpoint** for a DLI Flink Jar job.
   - Constraints on parameter optimization
     i. In the left navigation pane of the DLI console, choose **Global Configuration** > **Service Authorization**. On the page displayed, select **Tenant Administrator(Global service)** and click **Update**.
     ii. The bucket to which data is written must be an OBS bucket created by a main account.
   - Configuring **Restore Job from Checkpoint**
     i. Select **Auto Restart upon Exception**.
     ii. Select **Restore Job from Checkpoint** and set the **Checkpoint Path**.

     The checkpoint path is the same as that you set in JAR file code. The format is as follows:
     - ${bucket}/xxx/xxx/xxx
     - Example

       If the path in the JAR file is **obs://mybucket/jobs/checkpoint/jar-3**,

       Set **Checkpoint Path** to **mybucket/jobs/checkpoint/jar-3**.

     NOTE

     - The checkpoint path for each Flink Jar job must be unique. Otherwise, data cannot be restored.
     - DLI can access files in the checkpoint path only after DLI is authorized to access the OBS bucket.
3. Check whether the job is restored from the checkpoint.

14.1.20 Does a Flink JAR Job Support Configuration File Upload? How Do I Upload a Configuration File?

Configuration files can be uploaded for user-defined jobs (JAR).

1. Upload the configuration file to DLI through **Package Management**.
2. In the **Other Dependencies** area of the Flink JAR job, select the created DLI package.

3. Load the file through **ClassName.class.getClassLoader().getResource("userData/fileName")** in the code. In the file name, **fileName** indicates the name of the file to be accessed, and **ClassName** indicates the name of the class that needs to access the file.

14.1.21 Why Does the Submission Fail Due to Flink JAR File Conflict?

Symptom

The dependency of your Flink job conflicts with a built-in dependency of the DLI Flink platform. As a result, the job submission fails.

Solution

Delete your JAR file that is the same as an existing one of the DLI Flink platform.

14.1.22 Why Does a Flink Jar Job Fail to Access GaussDB(DWS) and a Message Is Displayed Indicating Too Many Client Connections?

Symptom

When a Flink Jar job is submitted to access GaussDB(DWS), an error message is displayed indicating that the job fails to be started. The job log contains the following error information:

```
FATAL:  Already too many clients, active/non-active/reserved: 5/508/3
```

Possible Causes

The number of GaussDB(DWS) database connections exceeds the upper limit. In the error information, the value of **non-active** indicates the number of idle connections. For example, if the value of **non-active** is 508, there are 508 idle connections.

Solution

Perform the following steps to solve the problem:

1. Log in to the GaussDB(DWS) command window and run the following SQL statement to release all idle (non-active) connections temporarily:
   ```
   SELECT PG_TERMINATE_BACKEND(pid) from pg_stat_activity WHERE state='idle';
   ```
2. Check whether the application actively releases the connections. If the application does not, optimize the code to release the connections.
3. On the GaussDB (DWS) management console, configure parameter **session_timeout**, which controls the timeout period of idle sessions. After an idle session's timeout period exceeds the specified value, the server automatically closes the connection.

   The default value of this parameter is **600** seconds. The value **0** indicates that the timeout limit is disabled. Do not set **session_timeout** to **0**.

The procedure for setting parameter **session_timeout** is as follows:

a. Log in to the GaussDB(DWS) management console.
b. In the navigation pane on the left, click **Clusters**.
c. In the cluster list, find the target cluster and click its name. The **Basic Information** page is displayed.
d. Click the **Parameter Modifications** tab and modify the value of parameter **session_timeout**. Then click **Save**.
e. In the **Modification Preview** dialog box, confirm the modification and click **Save**.

14.1.23 Why Is Error Message "Authentication failed" Displayed During Flink Jar Job Running?

Symptom

An exception occurred when a Flink Jar job is running. The following error information is displayed in the job log:

```
org.apache.flink.shaded.curator.org.apache.curator.ConnectionState - Authentication failed
```

Possible Causes

Service authorization is not configured for the account on the **Global Configuration** page. When the account is used to create a datasource connection to access external data, the access fails.

Solution

Step 1 Log in to the DLI management console. Choose **Global Configuration** > **Service Authorization** in the navigation pane.

Step 2 On the **Service Authorization** page, select all agency permissions.

Step 3 Click **Update**. If the message "Agency permissions updated successfully" is displayed, the modification is successful.

Step 4 After the authorization is complete, create a datasource connection and run the job again.

----End

14.1.24 Why Is Error Invalid OBS Bucket Name Reported After a Flink Job Submission Failed?

Symptom

The storage path of the Flink Jar job checkpoints was set to an OBS bucket. The job failed to be submitted, and an error message indicating an invalid OBS bucket name was displayed.

Cause Analysis

1. Check that the OBS bucket name is correct.
2. Check that the AK/SK has the required permission.
3. Set the dependency to **provided** to prevent JAR file conflicts.
4. Check that the **esdk-obs-java-3.1.3.jar** version is used.
5. Confirm that the cluster configuration is faulty.

Procedure

Step 1 Set the dependency to **provided**.

Step 2 Restart the **clusteragent** cluster after an upgrade to make the configuration take effect.

Step 3 Remove the OBS dependency. Otherwise, the checkpoints cannot be written to OBS.

----End

14.1.25 Why Does the Flink Submission Fail Due to Hadoop JAR File Conflict?

Symptom

Flink Job submission failed. The exception information is as follows:

```
Caused by: java.lang.RuntimeException: java.lang.ClassNotFoundException: Class
org.apache.hadoop.fs.obs.metrics.OBSAMetricsProvider not found
 at org.apache.hadoop.conf.Configuration.getClass(Configuration.java:2664)
 at org.apache.hadoop.conf.Configuration.getClass(Configuration.java:2688)
 ... 31 common frames omitted
 Caused by: java.lang.ClassNotFoundException: Class org.apache.hadoop.fs.obs.metrics.OBSAMetricsProvider
not found
 at org.apache.hadoop.conf.Configuration.getClassByName(Configuration.java:2568)
 at org.apache.hadoop.conf.Configuration.getClass(Configuration.java:2662)
 ... 32 common frames omitted
```

Cause Analysis

Flink JAR files conflicted. The submitted Flink JAR file conflicted with the HDFS JAR file of the DLI cluster.

Procedure

Step 1 Configure **hadoop-hdfs** in the POM file as follows:

```
<dependency>
<groupId>org.apache.hadoop</groupId>
<artifactId>hadoop-hdfs</artifactId>
<version>${hadoop.version}</version>
<scope> provided </scope>
</dependency>
```

Alternatively, use the **exclusions** tag to exclude the association.

Step 2 To use HDFS configuration files, change **core-site.xml**, **hdfs-site.xml**, and **yarn-site.xml** to **mrs-core-site.xml**, **mrs-hdfs-site.xml** and **mrs-hbase-site.xml**, respectively.

```
conf.addResource(HBaseUtil.class.getClassLoader().getResourceAsStream("mrs-core-site.xml"), false);
conf.addResource(HBaseUtil.class.getClassLoader().getResourceAsStream("mrs-hdfs-site.xml"), false);
conf.addResource(HBaseUtil.class.getClassLoader().getResourceAsStream("mrs-hbase-site.xml"), false);
```

----End

14.1.26 How Do I Connect a Flink jar Job to SASL_SSL?

You can use Flink Jar to connect to Kafka with SASL SSL authentication enabled.

14.1.27 How Do I Optimize Performance of a Flink Job?

Basic Concepts and Job Monitoring

- Data Stacking in a Consumer Group

  The accumulated data of a consumer group can be calculated by the following formula: Total amount of data to be consumed by the consumer group = Offset of the latest data – Offset of the data submitted to the consumer group

  If your Flink job is connected to the Kafka premium edition, you can view the customer group on the Cloud Eye console. To view consumer available messages, choose **Cloud Service Monitoring** > **Distributed Message Service** form the navigation pane. On the displayed page, select **Kafka Premium** and click the **Consumer Groups** tab. Click the Kafka instance name and select the target consumer group.

- Back Pressure Status

  Back pressure status is working load status of an operator. The back pressure is determined by the ratio of threads blocked in the output buffer to the total taskManager threads. This ratio is calculated by periodically sampling of the taskManager thread stack. By default, if the ratio is less than 0.1, the back pressure status is OK. If the ratio ranges from 0.1 to 0.5, the backpressure status is LOW. If the ratio exceeds 0.5, the backpressure status is HIGH.

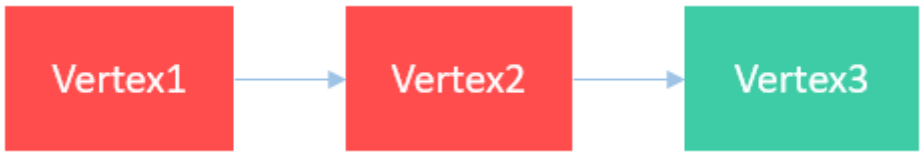
- Delay

  Delay indicates the duration from the time when source data starts being processed to the time when data reaches the current operator. The data source periodically sends a LatencyMarker (current timestamp). After receiving the LatencyMarker, the downstream operator subtracts the timestamp from the current time to calculate the duration. You can view the back pressure status and delay of an operator on the Flink UI or in the task list of a job. Generally, high back pressure and delay occur in pairs.

Performance Analysis

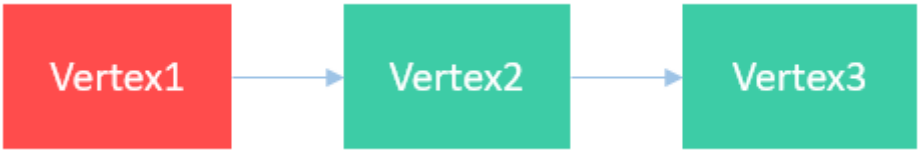
Due to Flink back pressure, the data source consumption rate can be lower than the production rate when performance of a Flink job is low. As a result, data is stacked in a Kafka consumer group. In this case, you can use back pressure and delay of the operator to find its performance bottleneck.

- The following figure shows that the back pressure of the last operator (sink) of the job is normal (green), and the back pressure of the previous two operators is high (red).



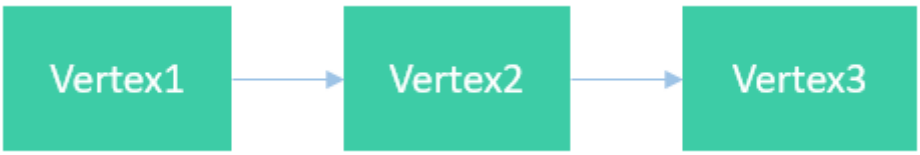
  In this scenario, the performance bottleneck is the sink and the optimization is specific to the data source. For example, for the JDBC data source, you can adjust the write batch using **connector.write.flush.max-rows** and JDBC rewriting parameter **rewriteBatchedStatements=true** to optimize the performance.

- The following figure shows a scenario where the back pressure of the last second operator is normal.



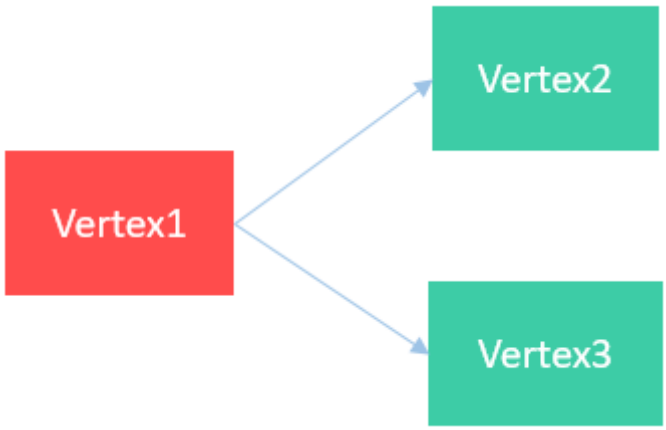
  In this scenario, the performance bottleneck is the Vertex2 operator. You can view the description about the function of the operator for further optimization.

- The back pressure of all operators is normal, but data is stacked.



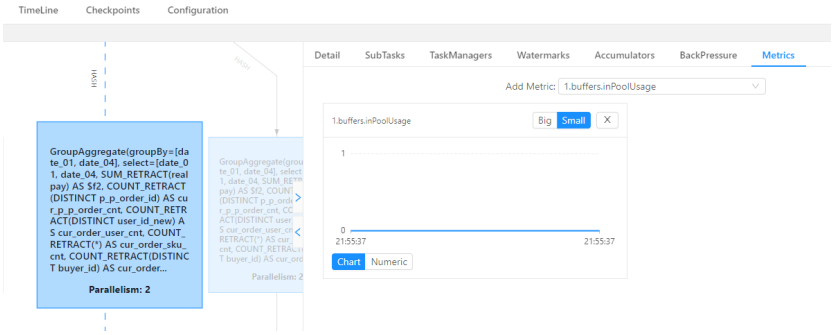
  In this scenario, the performance bottleneck is the source, and the performance is mainly affected by the data read speed. In this case, you can increase the number of Kafka partitions and the number of concurrent sources to solve the problem.

- The following figure shows that the back pressure of an operator is high, and its subsequent concurrent operators do not have back pressure.



In this scenario, the performance bottleneck is Vertex2 or Vertex3. To find out the specific bottleneck operator, enable inPoolUsage monitoring on the Flink UI page. If the inPoolUsage for operator concurrency is 100% for a long time, the corresponding operator is likely to be the performance bottleneck. In this case, you check the operator for further optimization.

Figure 14-1 inPoolUsage monitoring



Performance Tuning

- Rocksdb state tuning

  Top N sorting, window aggregate calculation, and stream-stream join involve a large number of status operations. You can optimize the performance of state operations to improve the overall performance. You can try any of the following optimization methods:

  - Increase the state operation memory and reduce the disk I/O.
    - Increase the number of CU resources in a single slot.
    - Set optimization parameters:
      - taskmanager.memory.managed.fraction=xx
      - state.backend.rocksdb.block.cache-size=xx
      - state.backend.rocksdb.writebuffer.size=xx
  - Enable the micro-batch mode to avoid frequent state operations.

    Set the following parameters:
    - table.exec.mini-batch.enabled=true
    - table.exec.mini-batch.allow-latency=xx
    - table.exec.mini-batch.size=xx
  - Use ultra-high I/O local disks to accelerate disk operations.
- Group aggregation tuning

  The data skew problem is solved by Local-Global that divides a group aggregation into two stages: doing local aggregation in upstream first, and then global aggregation in downstream. To enable Local-global aggregation, set optimization parameter: **table.optimizer.aggphase-strategy=TWO_PHASE**
- Tuning count distinct

  - If the associated keys of count distinct are sparse, using Local-Globa cannot solve the problem of SPOF. In this case, you can configure the following parameters to optimize bucket splitting.
    - table.optimizer.distinct-agg.split.enabled=true
    - table.optimizer.distinct-agg.split.bucket-num=xx
  - Replace CASE WHEN with FILTER:

    For example:

    ```
    COUNT(DISTINCT CASE WHEN flag IN ('android', 'iphone')THEN user_id ELSE NULL END) AS app_uv
    ```

    Can be changed to:

    ```
    COUNT(DISTINCT user_id) FILTER(WHERE flag IN ('android', 'iphone')) AS app_uv
    ```

- Optimizing dimension table join

  The dimension table in joined with the key of each record in the left table. The matched in the cache is performed first. If no match is found, the remotely obtained data is used for matching. The optimization is as follows:
  - Increase the JVM memory and the number of cached records.
  - Set indexes for the dimension table to speed up query.

14.1.28 How Do I Write Data to Different Elasticsearch Clusters in a Flink Job?

Add the following SQL statements to the Flink job:

```
create source stream ssource(xx);
create sink stream es1(xx) with (xx);
create sink stream es2(xx) with (xx);
insert into es1 select * from ssource;
insert into es2 select * from ssource;
```

14.1.29 How Do I Prevent Data Loss After Flink Job Restart?

The DLI Flink checkpoint/savepoint mechanism is complete and reliable. You can use this mechanism to prevent data loss when a job is manually restarted or restarted due to an exception.

- To prevent data loss caused by job restart due to system faults, perform the following operations:
  - For Flink SQL jobs, select **Enable Checkpointing** and set a proper checkpoint interval that allows for the impact on service performance and the exception recovery duration. Select **Auto Restart upon Exception** and **Restore Job from Checkpoint**. After the configuration, if a job is restarted abnormally, the internal state and consumption position will be restored from the latest checkpoint file to ensure no data loss and accurate and consistent semantics of the internal state such as aggregation operators. In addition, to ensure that data is not duplicated, use a database or file system with a primary key as the data source. Otherwise, add deduplication logic (data generated from the latest successful checkpoint to the time exception occurred will be repeatedly consumed) for downstream processes.
  - For Flink Jar jobs, you need to enable the checkpoint in the code. If a user-defined state needs to be saved, you need to implement the

ListCheckpointed API, set a unique ID for each operator. In the job configuration, select **Restore Job from Checkpoint** and configure the checkpoint path.

NOTE

Flink checkpointing ensures that the internal state data is accurate and consistent. However, for custom Source/Sink or stateful operators, you need to implement the ListCheckpointed API to ensure the reliability of service data.

- To prevent data loss after a job is manually restarted due to service modification, perform the following operations:
  - For jobs without internal states, you can set the start time or consumption position of the Kafka data source to a time before the job stops.
  - For jobs with internal states, you can select **Trigger Savepoint** when stopping the job. Enable **Restore Savepoint** when you start the job again. The job will restore the consumption position and state from the selected savepoint file. The generation mechanism and format of Flink checkpoints are the same as those of savepoints. You can go to the Flink job list and choose **More** > **Import Savepoint** in the **Operation** column of a Flink job to import the latest checkpoint in OBS and restore the job from it.

14.1.30 How Do I Locate a Flink Job Submission Error?

1. On the Flink job management page, hover the cursor on the status of the job that fails to be submitted to view the brief information about the failure.

   The possible causes are as follows:

   - Insufficient CUs: Increase the number of CUs of the queue.
   - Failed to generate the JAR file: Check the SQL syntax and UDFs.
2. If you cannot locate the fault or the call stack is incorrect, click the job name to go to the job details page and click the **Commit Logs** tab to view the job submission logs.

14.1.31 How Do I Locate a Flink Job Running Error?

1. On the Flink job management, click **Edit** in the **Operation** column of the target job. On the displayed page, check whether **Save Job Log** in the **Running Parameters** tab is enabled.
   - If the function is enabled, go to **3**.
   - If the function is disabled, running logs will not be dumped to an OBS bucket. In this case, perform **2** to save job logs.
2. On the job running page, select **Save Job Log** and specify an OBS bucket for storing the logs. Click **Start** to run the job again. After the executed is complete, perform **3** and subsequent steps.
3. In the Flink job list, click the job name. On the displayed job details page, click the **Run Log** tab.
4. Click **view OBS Bucket** to obtain the complete run logs of the job.
5. Download the latest **jobmanager.log** file, search for the keyword **RUNNING to FAILED**, and determine the failure cause based on the errors in the context.

6. If the information in the **jobmanager.log** file is insufficient for locating the fault, find the corresponding **taskmanager.log** file in the run logs and search for the keyword **RUNNING to FAILED** to confirm the failure cause.

14.1.32 How Do I Know Whether a Flink Job Can Be Restored from a Checkpoint After Being Restarted?

Check the following operations:

- Adjusting or adding optimization parameters or the number of concurrent threads of a job, or modifying Flink SQL statements or a Flink Jar job: The job cannot be restored from the checkpoint.
- Modifying the number of CUs occupied by a TaskManager: The job can be restored from the checkpoint.

14.1.33 Why Does DIS Stream Not Exist During Job Semantic Check?

To rectify this fault, perform the following steps:

1. Log in to the DIS management console. In the navigation pane, choose **Stream Management**. View the Flink job SQL statements to check whether the DIS stream exists.
2. If the DIS stream was not created, create a DIS stream by referring to "Creating a DIS Stream" in the Data Ingestion Service User Guide.

   Ensure that the created DIS stream and Flink job are in the same region.
3. If a DIS stream has been created, check whether the DIS stream and the Flink job are in the same region.

14.1.34 Why Is the OBS Bucket Selected for Job Not Authorized?

If the OBS bucket selected for a job is not authorized, perform the following steps:

Step 1 On the DLI management console, choose **Global Configuration** > **Service Authorization** and select the **Tenant Administrator (Global service)** permission. Update the permission settings.

Step 2 In the navigation pane, choose **Job Management** > **Flink Jobs**.

Step 3 In the row where the target job resides, click **Edit** under **Operation** to switch to the **Edit** page.

Step 4 Configure parameters under **Running Parameters** on the **Edit** page.

1. Select **Enable Checkpointing** or **Save Job Log**.
2. Specify **OBS Bucket**.
3. Select **Authorize OBS**.

----End

14.1.35 Why Are Logs Not Written to the OBS Bucket After a DLI Flink Job Fails to Be Submitted for Running?

Mode for storing generated job logs when a DLI Flink job fails to be submitted or executed. The options are as follows:

- If the submission fails, a submission log is generated only in the **submit-client** directory.
- You can view the logs generated within 1 minute when the job fails to be executed on the management console.

  Choose **Job Management** > **Flink Jobs**, click the target job name to go to the job details page, and click **Run Log** to view real-time logs.
- If the running fails and exceeds 1 minute (the log dump period is 1 minute), run logs are generated in the **application**_*xx* directory.

Flink dependencies have been built in the DLI server and security hardening has been performed based on the open-source community version. To avoid dependency package compatibility issues or log output and dump issues, be careful to exclude the following files when packaging:

- Built-in dependencies (or set the package dependency scope to "provided")
- Log configuration files (example, log4j.properties/logback.xml)
- JAR file for log output implementation (example, log4j).

On this basis, the **taskmanager.log** file rolls as the log file size and time change.

14.1.36 Why Is Information Displayed on the FlinkUI/Spark UI Page Incomplete?

Symptom

The Flink/Spark UI was displayed with incomplete information.

Possible Causes

When the queue is used to run a job, the system releases the cluster and takes about 10 minutes to create a new one. Accessing the Flink UI before completion of the creation will empty the project ID in the cache. As a result, the UI cannot be displayed. The possible cause is that the cluster was not created.

Solution

Change the queue to dedicated, so that the cluster will not be released when the queue is idle. Alternatively, submit a job, wait for a while, and then access FlinkUI.

14.1.37 Why Is the Flink Job Abnormal Due to Heartbeat Timeout Between JobManager and TaskManager?

Symptom

JobManager and TaskManager heartbeats timed out. As a result, the Flink job is abnormal.

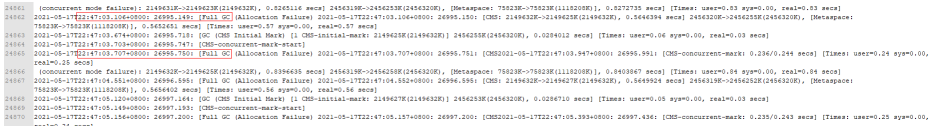
Figure 14-2 Error information

```
Jobmanager. log

2021-05-17 22:44:37.312 INFO  [70] org.apache.flink.runtime.checkpoint.CheckpointCoordinator   - Triggering checkpoint 223 @ 1621262677310 for job 00d04eb6e7147e59f2d2877bdb48ce4d.
22:46:01.729 jobmanager displays heartbeat timeout
2021-05-17 22:46:01.729 INFO  [3659] org.apache.flink.runtime.executiongraph.ExecutionGraph   - Map -> Map (4/4) (9e6b13c7ab5d3637062f1697014b4eff) switched from RUNNING to FAILED.
java.util.concurrent.TimeoutException: Heartbeat of TaskManager with id container_1619690508608_1067_01_000003 timed out.
        at org.apache.flink.runtime.jobmaster.JobMaster$TaskManagerHeartbeatListener.notifyHeartbeatTimeout(JobMaster.java:1642)
        at org.apache.flink.runtime.heartbeat.HeartbeatManagerImpl$HeartbeatMonitor.run(HeartbeatManagerImpl.java:335)

container_1619690508608_1067_01_000003

the zookeeper client times out
2021-05-17 22:45:44.078 INFO  [85] org.apache.zookeeper.ClientCnxn   - Client session timed out, have not heard from server in 64331ms for sessionid 0x18000013858becca, closing socket connection and attempting reconnect
2021-05-17 22:46:39.547 INFO  [85] org.apache.zookeeper.client.FourLetterWordMain   - connecting to node-master1fmxz.b1039268-c9c0-4c82-b21f-b1252b0f186f.com 2181
2021-05-17 22:47:00.358 INFO  [22] org.apache.flink.shaded.zookeeper.org.apache.zookeeper.ClientCnxn - Unable to read additional data from server sessionid 0x2005c1aeb4c0021, likely server has closed socket, closing socket connection and attempting reconnect
2021-05-17 22:47:11.223 WARN  [85] org.apache.zookeeper.ClientCnxn   - SASL configuration failed: javax.security.auth.login.LoginException: No JAAS configuration section named 'Client' was found in specified JAAS configuration file: '/tmp/jaas-2036661896892296518.conf'. Will continue connection to Zookeeper server without SASL authentication, if Zookeeper server allows it.
```

Possible Causes

1. Check whether the network is intermittently disconnected and whether the cluster load is high.
2. If Full GC occurs frequently, check the code to determine whether memory leakage occurs.

Figure 14-3 Full GC



Handling Procedure

- If Full GC occurs frequently, check the code to determine whether memory leakage occurs.
- Allocate more resources for a single TaskManager.
- Contact technical support to modify the cluster heartbeat configuration.

14.1.38 Why Is Error "Timeout expired while fetching topic metadata" Repeatedly Reported in Flink JobManager Logs?

1. Test address connectivity.
2. If the network is unreachable, rectify the network connection first. Ensure that the network connection between the DLI queue and the external data source is normal.

14.2 Problems Related to SQL Jobs

14.2.1 SQL Jobs

Can I Create Temporary Tables on DLI?

A temporary table is used to store intermediate results. When a transaction or session ends, the data in the temporary table can be automatically deleted. For example, in MySQL, you can use **create temporary table...** to create a temporary table. After a transaction or session ends, the table data is automatically deleted. Does DLI Support This Function?

Currently, **you cannot create temporary tables on DLI**. You can create a table by using SQL statements.

Can I Connect to DLI Locally? Is a Remote Connection Tool Supported?

Currently DLI can only be accessed through a browser. You must submit jobs on the console.

Will a DLI SQL Job Be Killed If It Has Been Running for More Than 12 Hours?

By default, SQL jobs that have been running for more than 12 hours will be canceled to ensure stability of queues.

You can use the **dli.sql.job.timeout** parameter (unit: second) to configure the timeout interval.

Does DLI Support Local Testing of Spark Jobs?

Currently, DLI does not support local testing of Spark jobs. You can install the DLI Livy tool and use its interactive sessions to debug Spark jobs.

Can I Delete a Row of Data from an OBS Table or DLI Table?

Deleting a row of data from an OBS table or DLI table is not allowed.

14.2.2 How Do I Merge Small Files?

If a large number of small files are generated during SQL execution, job execution and table query will take a long time. In this case, you should merge small files.

1. Set the configuration item as follows:

   spark.sql.shuffle.partitions = Number of partitions (number of the generated small files in this case)

2. Execute the following SQL statements:

```
INSERT OVERWRITE TABLE tablename
select  * FROM  tablename distribute by rand()
```

14.2.3 How Do I Specify an OBS Path When Creating an OBS Table?

Scenario

When creating an OBS table, you must specify a table path in the database. The path format is as follows: obs://xxx/database name/table name.

Correct Example

```
CREATE TABLE `di_seller_task_activity_30d` (`user_id` STRING COMMENT' user ID...) SORTED as parquet
LOCATION 'obs://akc-bigdata/akdc.db/di_seller_task_activity_30d'
```

Incorrect Example

```
CREATE TABLE `di_seller_task_activity_30d` (`user_id` STRING COMMENT' user ID...) SORTED as parquet
LOCATION 'obs://akc-bigdata/akdc.db'
```

NOTE

If the specified path is **akdc.db**, data in this path will be cleared when the **insert overwrite** statement is executed.

14.2.4 How Do I Create a Table Using JSON Data in an OBS Bucket?

DLI allows you to associate JSON data in an OBS bucket to create tables in asynchronous mode.

The statement for creating the table is as follows:

```
create table tb1 using json options(path 'obs://....')
```

14.2.5 How Do I Set Local Variables in SQL Statements?

You can use the **where** condition statement in the **select** statement to filter data. For example:

```
select * from table where part = '202012'
```

14.2.6 How Can I Use the count Function to Perform Aggregation?

The correct method for using the count function to perform aggregation is as follows:

```
SELECT
  http_method,
  count(http_method)
FROM
  apigateway
WHERE
  service_id = 'ecs' Group BY http_method
```

Or

```
SELECT
  http_method
FROM
```

```
  apigateway
WHERE
  service_id = 'ecs' DISTRIBUTE BY http_method
```

If an incorrect method is used, an error will be reported.

```
SELECT
  http_method,
  count(http_method)
FROM
  apigateway
WHERE
  service_id = 'ecs' DISTRIBUTE BY http_method
```

14.2.7 How Do I Synchronize DLI Table Data from One Region to Another?

You can use the cross-region replication function of OBS. The procedure is as follows:

1. Export the DLI table data in region 1 to the user-defined OBS bucket.
2. Use the OBS cross-region replication function to replicate data to the OBS bucket in region 2.
3. Import or use the corresponding data as required.

14.2.8 How Do I Insert Table Data into Specific Fields of a Table Using a SQL Job?

Currently, DLI does not allow you to insert table data into specific fields. To insert table data, you must insert data of all table fields at a time.

14.2.9 Why Is Error "path obs://xxx already exists" Reported When Data Is Exported to OBS?

Create an OBS directory with a unique name. Alternatively, you can manually delete the existing OBS directory and submit the job again. However, exercise caution when deleting the existing OBS directory because the operation will delete all data in the directory.

14.2.10 Why Is Error "SQL_ANALYSIS_ERROR: Reference 't.id' is ambiguous, could be: t.id, t.id.;" Displayed When Two Tables Are Joined?

This message indicates that the two tables to be joined contain the same column, but the owner of the column is not specified when the command is executed.

For example, tables tb1 and tb2 contain the **id** field.

Incorrect command:

```
select id from tb1 join tb2;
```

Correct command:

```
select tb1.id from tb1 join tb2;
```

14.2.11 Why Is Error "The current account does not have permission to perform this operation,the current account was restricted. Restricted for no budget." Reported when a SQL Statement Is Executed?

Check whether the account is in arrears. If yes, recharge the account.

If the error persists, log out and log in again.

14.2.12 Why Is Error "There should be at least one partition pruning predicate on partitioned table XX.YYY" Reported When a Query Statement Is Executed?

- Cause Analysis

  When you query the partitioned table **XX.YYY**, the partition column is not specified in the search criteria.

  A partitioned table can be queried only when the query condition contains at least one partition column.

- Solution

  Query a partitioned table by referring to the following example:

  Assume that **partitionedTable** is a partitioned table and **partitionedColumn** is a partition column. The query statement is as follows:

  ```
  SELECT * FROM partitionedTable WHERE partitionedColumn  = XXX
  ```

14.2.13 Why Is Error "IllegalArgumentException: Buffer size too small. size" Reported When Data Is Loaded to an OBS Foreign Table?

Symptom

The following error message is displayed when the LOAD DATA command is executed by a Spark SQL job to import data to a DLI table:

```
error.DLI.0001: IllegalArgumentException: Buffer size too small. size = 262144 needed = 2272881
```

In some cases ,the following error message is displayed:

```
error.DLI.0999: InvalidProtocolBufferException: EOF in compressed stream footer position: 3 length: 479 range: 0 offset: 3 limit: 479 range 0 = 0 to 479 while trying to read 143805 bytes
```

Possible Causes

The data volume of the file to be imported is large and the value of **spark.sql.shuffle.partitions** is too large. As a result, the cache size is insufficient.

Solution

Decrease the **spark.sql.shuffle.partitions** value. To set this parameter, perform the following steps:

1. Log in to the DLI management console and choose **Job Management** > **SQL Jobs**. In the **Operation** column of the target SQL job, click **Edit** to go to the **SQL Editor** page.
2. On the displayed page, click **Set Property** and set the parameter.
3. Execute the job again.

14.2.14 Why Is Error "DLI.0002 FileNotFoundException" Reported During SQL Job Running?

Symptom

An error is reported during SQL job execution:

```
Please contact DLI service. DLI.0002: FileNotFoundException: getFileStatus on obs://xxx: status [404]
```

Solution

Check whether there is another job that has deleted table information.

DLI does not allow multiple jobs to read and write the same table at the same time. Otherwise, job conflicts may occur and the jobs fail.

14.2.15 Why Is a Schema Parsing Error Reported When I Create a Hive Table Using CTAS?

Currently, DLI supports the Hive syntax for creating tables of the TEXTFILE, SEQUENCEFILE, RCFILE, ORC, AVRO, and PARQUET file types. If the file format specified for creating a table in the CTAS is AVRO and digits are directly used as the input of the query statement (SELECT), for example, if the query is **CREATE TABLE tb_avro STORED AS AVRO AS SELECT 1**, a schema parsing exception is reported.

If the column name is not specified, the content after SELECT is used as both the column name and inserted value. The column name of the AVRO table cannot be a digit. Otherwise, an error will be reported, indicating that the schema fails to be parsed.

Solution: You can use **CREATE TABLE tb_avro STORED AS AVRO AS SELECT 1 AS colName** to specify the column name or set the storage format to a format other than AVRO.

14.2.16 Why Is Error "org.apache.hadoop.fs.obs.OBSIOException" Reported When I Run DLI SQL Scripts on DataArts Studio?

Symptom

When you run a DLI SQL script on DataArts Studio, the log shows that the statements fail to be executed. The error information is as follows:

```
DLI.0999: RuntimeException: org.apache.hadoop.fs.obs.OBSIOException: initializing on obs://xxx.csv: status [-1] - request id
[null] - error code [null] - error message [null] - trace :com.obs.services.exception.ObsException: OBS servcie Error Message. Request Error:
```

```
...
Cause by: ObsException: com.obs.services.exception.ObsException: OBSs servcie Error Message. Request
Error: java.net.UnknownHostException: xxx: Name or service not known
```

Possible Causes

When you execute a DLI SQL script for the first time, you did not agree to the privacy agreement on the DLI console. As a result, the error is reported when the SQL script is executed on DataArts Studio.

Solution

1. Log in to the DLI console, click **SQL Editor** from the navigation pane. On the displayed page, enter an SQL statement in the editing window, for example, **select 1**.
2. In the displayed **Privacy Agreement** dialog box, agree to the terms.

   NOTE

   You only need to agree to the privacy agreement when it is your first time to execute the statements.
3. Run the DLI SQL script on DataArts Studio again. The script will run properly.

14.2.17 Why Is Error "UQUERY_CONNECTOR_0001:Invoke DLI service api failed" Reported in the Job Log When I Use CDM to Migrate Data to DLI?

Symptom

After the migration job is submitted, the following error information is displayed in the log:

```
org.apache.sqoop.common.SqoopException: UQUERY_CONNECTOR_0001:Invoke DLI service api failed,
failed reason is %s.
at org.apache.sqoop.connector.uquery.intf.impl.UQueryWriter.close(UQueryWriter.java:42)
at org.apache.sqoop.connector.uquery.processor.Dataconsumer.run(Dataconsumer.java:217)
at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
at java.lang.Thread.run(Thread.java:748)
```

Possible Causes

When you create a migration job to DLI on the CDM console, you set **Resource Queue** to a DLI queue for general purpose. It should be a queue for SQL.

Solution

1. On the DLI management console and click **Queue Management** in the navigation pane on the left. On the **Queue Management** page, check whether there are SQL queues.
   - If there are, go to **3**.
   - If there are no SQL queues, go to **2** to buy an SQL queue.
2. Click **Buy Queue** to create a queue. Set **Type** to **For SQL**, set other parameters required, and click **Buy**.

3. Go back to the CDM console and create a data migration job. Set **Resource Queue** to the created DLI SQL queue.
4. Submit the migration job and view the job execution logs.

14.2.18 Why Is Error "File not Found" Reported When I Access a SQL Job?

Symptom

Error message "File not Found" is displayed when a SQL job is accessed.

Solution

Generally, the file cannot be found due to a read/write conflict. Check whether a job is overwriting the data when the error occurs.

14.2.19 Why Is Error "DLI.0003: AccessControlException XXX" Reported When I Access a SQL Job?

Symptom

Error message "DLI.0003: AccessControlException XXX" is reported when a SQL job is accessed.

Solution

View the OBS bucket in the AccessControlException and check whether you are using an account that has the permission to access the bucket.

14.2.20 Why Is Error "DLI.0001: org.apache.hadoop.security.AccessControlException: verifyBucketExists on {{bucket name}}: status [403]" Reported When I Access a SQL Job?

Symptom

Error message "DLI.0001: org.apache.hadoop.security.AccessControlException: verifyBucketExists on {{bucket name}}: status [403]" is reported when a SQL job is Accessed.

Solution

The current account does not have the permission to access the OBS bucket where the foreign table is located. Obtain the OBS permission and perform the query again.

14.2.21 Why Is Error "The current account does not have permission to perform this operation,the current account was restricted. Restricted for no budget" Reported During SQL Statement Execution? Restricted for no budget.

Symptom

Error message "The current account does not have permission to perform this operation,the current account was restricted." is reported during SQL statement execution.

Solution

Check whether your account is in arrears. If it is, renew your account and try again.

If the error persists after renewal, log out and log in again.

14.2.22 How Do I Troubleshoot Slow SQL Jobs?

If the job runs slowly, perform the following steps to find the causes and rectify the fault:

Possible Cause 1: Full GC

Check whether the problem is caused by FullGC.

1. Log in to the DLI console. In the navigation pane, choose **Job Management** > **SQL Jobs**.
2. On the **SQL Jobs** page, locate the row that contains the target job and click **More** > **View Log** in the **Operation** column.
3. Obtain the folder of the archived logs in the OBS directory. The details are as follows:
   - Spark SQL jobs:

     Locate the log folder whose name contains **driver** or **container**_ *xxx* _**000001**.
   - Spark Jar jobs:

     The archive log folder of a Spark Jar job starts with **batch**.
4. Go to the archive log file directory and download the **gc.log.* log** file.
5. Open the downloaded **gc.log.* log** file, search for keyword **Full GC**, and check whether time records in the file are continuous and Full GC information is recorded repeatedly.

Cause locating and solution

Cause 1: There are too many small files in a table.

1. Log in to the DLI console and go to the SQL editor page. On the SQL Editor page, select the queue and database of the faulty job.

2. Run the following statement to check the number of files in the table and specify the *table name*.

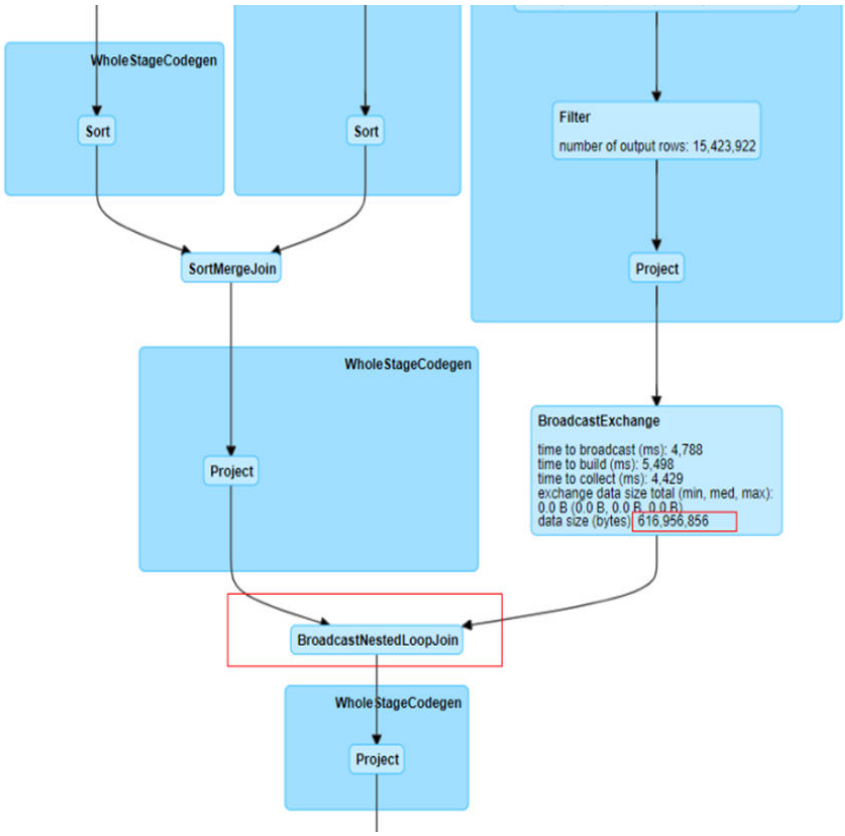
```
select count(distinct fn)  FROM
(select input_file_name() as fn from table name) a
```

3. If there are too many small files, rectify the fault by referring to **How Do I Merge Small Files?**.

Cause 2: There is a broadcast table.

1. Log in to the DLI console. In the navigation pane, choose **Job Management** > **SQL Jobs**.
2. On the **SQL Jobs** page, locate the row that contains the target job and click to view the job details and obtain the job ID.
3. In the **Operation** column of the job, click **Spark UI**.
4. On the displayed page, choose **SQL** from the menu bar. Click the hyperlink in the **Description** column of the row that contains the job ID.
5. View the DAG of the job to check whether the BroadcastNestedLoopJoin node exists.

Figure 14-4 DAG

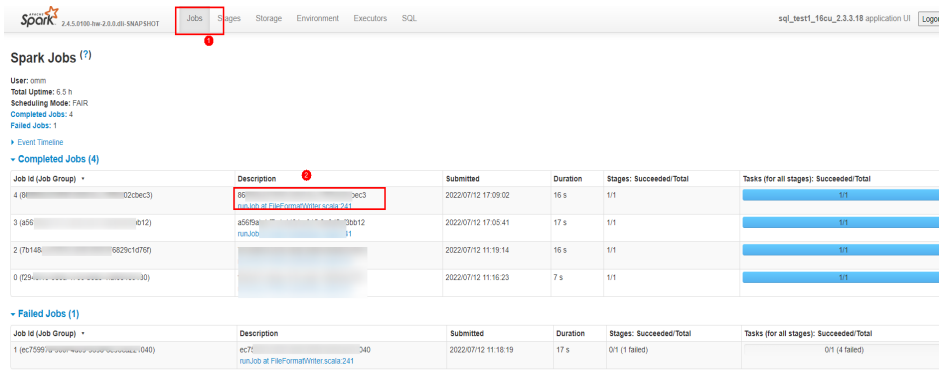


6. If the BroadcastNestedLoopJoin node exists, refer to **Why Does a SQL Job That Has Join Operations Stay in the Running State?** to rectify the fault.

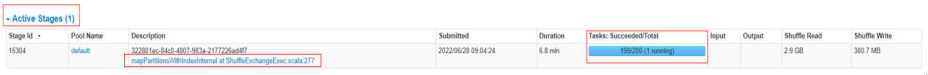
Possible Cause 2: Data Skew

Check whether the problem is caused by data skew.

1. Log in to the DLI console. In the navigation pane, choose **Job Management** > **SQL Jobs**.
2. On the **SQL Jobs** page, locate the row that contains the target job and click to view the job details and obtain the job ID.
3. In the **Operation** column of the job, click **Spark UI**.
4. On the displayed page, choose **SQL** from the menu bar. Click the hyperlink in the **Description** column of the row that contains the job ID.



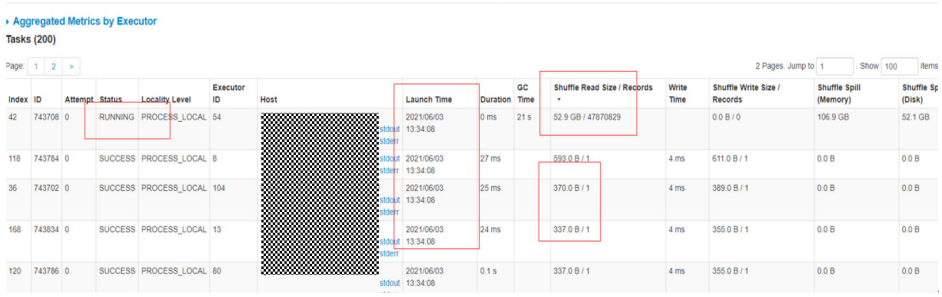
5. View the running status of the current stage in the Active Stage table on the displayed page. Click the hyperlink in the **Description** column.



6. View the **Launch Time** and **Duration** of each task.
7. Click **Duration** to sort tasks. Check whether the overall job duration is prolonged because a task has taken a long time.

   According to **Figure 14-5**, when data skew occurs, the data volume of shuffle reads of a task is much greater than that of other tasks.

   Figure 14-5 Data skew



Cause locating and solution

Shuffle data skew is caused by unbalanced number of key values in join.

1. Perform **group by** and **count** on a join to collect statistics on the number of key values of each join. The following is an example:

   Join table **lefttbl** and table **righttbl**. **num** in the **lefttbl** table is the key value of the join. You can perform **group by** and **count** on **lefttbl.num**.

```
SELECT * FROM lefttbl a LEFT join righttbl b on a.num = b.int2;
SELECT count(1) as count,num from lefttbl  group by lefttbl.num ORDER BY count desc;
```

2. Use **concat(cast(round(rand() * 999999999) as string)** to generate a random number for each key value.
3. If the skew is serious and random numbers cannot be generated, see **How Do I Eliminate Data Skew by Configuring AE Parameters?**

14.2.23 How Do I View DLI SQL Logs?

Scenario

You can view SQL job logs for routine O&M.

Procedure

1. Obtain the ID of the DLI job executed on the DataArts Studio console.

   Figure 14-6 Job ID

   ```
    ) t;
   [2021/03/19 10:02:59 GMT+0800] [INFO] dli.sql.badRecordsPath=obs://dlf-log-060f26783a00109c2f7ac00d662b
   [2021/03/19 10:03:00 GMT+0800] [INFO] DLI job id is:3d249de0-96c3-46d7-a788-adde1207376a
   [2021/03/19 10:03:00 GMT+0800] [INFO] Spark UI path of DLI SQL Job [dm_trfc_prd_high_commission_info] is [ht
   [2021/03/19 10:03:00 GMT+0800] [INFO] submit job success. Waiting for job execution to complete...
   [2021/03/19 10:04:00 GMT+0800] [INFO] DLI sql execute success
   ```

2. On the DLI console, choose **Job Management** > **SQL Jobs**.
3. On the **SQL Jobs** page, enter the job ID.
4. In the **Operation** column of the target job, choose **More** > **View Log** and download the logs to your PC.
5. Search for job ID in the downloaded logs to view the execution logs.

14.2.24 How Do I View SQL Execution Records?

Scenario

You can view the job execution records when a job is running.

Procedure

1. Log in to the DLI management console.
2. In the navigation pane on the left, choose **Job Management** > **SQL Jobs**.
3. Enter a job ID or statement to search for a job.

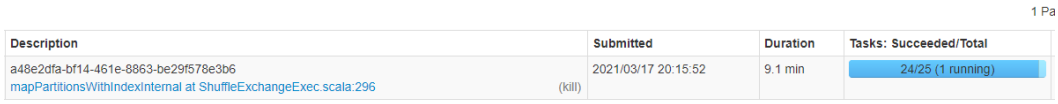
14.2.25 How Do I Eliminate Data Skew by Configuring AE Parameters?

Scenario

If the execution of an SQL statement takes a long time, you need to access the Spark UI to check the execution status.

If data skew occurs, the running time of a stage exceeds 20 minutes and only one task is running.

Figure 14-7 Data skew example



Procedure

1. Log in to the DLI management console. Choose **Job Management** > **SQL Jobs** in the navigation pane. On the displayed page, locate the job you want to modify and click **Edit** in the **Operation** column to switch to the **SQL Editor** page.
2. On the **SQL editor** page, click **Set Property** and add the following Spark parameters through the **Settings** pane:

   The string followed by the colons (:) are the configuration parameters, and the strings following the colons are the values.

   ```
   spark.sql.enableToString:false
   spark.sql.adaptive.join.enabled:true
   spark.sql.adaptive.enabled:true
   spark.sql.adaptive.skewedJoin.enabled:true
   spark.sql.adaptive.enableToString:false
   spark.sql.adaptive.skewedPartitionMaxSplits:10
   ```

   NOTE

   spark.sql.adaptive.skewedPartitionMaxSplits indicates the maximum number of tasks for processing a skewed partition. The default value is **5**, and the maximum value is **10**. This parameter is optional.
3. Click **Execute** to run the job again.

14.2.26 What Can I Do If a Table Cannot Be Queried on the DLI Console?

Symptom

A DLI table exists but cannot be queried on the DLI console.

Possible Causes

If a table exists but cannot be queried, there is a high probability that the current user does not have the permission to query or operate the table.

Solution

Contact the user who creates the table and obtain the required permissions. To assign permissions, perform the following steps:

1. Log in to the DLI management console as the user who creates the table. Choose **Data Management** > **Databases and Tables** form the navigation pane on the left.

2. Click the database name. The table management page is displayed. In the **Operation** column of the target table, click **Permissions**. The table permission management page is displayed.
3. Click **Set Permission**. In the displayed dialog box, set **Authorization Object** to **User**, set **Username** to the name of the user that requires the permission, and select the required permissions. For example, **Select Table** and **Insert** permissions.
4. Click **OK**.
5. Log in to the DLI console as the user that has been granted permission and check whether the table can be queried.

14.2.27 The Compression Ratio of OBS Tables Is Too High

A high compression ratio of OBS tables in the Parquet or ORC format (for example, a compression ratio of 5 or higher compared with text compression) will lead to large data volumes to be processed by a single task. In this case, you are advised to set **dli.sql.files.maxPartitionBytes** to **33554432** (default: **134217728**) in the **conf** field in the **submit-job** request body to reduce the data to be processed per task.

14.2.28 How Can I Avoid Garbled Characters Caused by Inconsistent Character Codes?

DLI supports only UTF-8-encoded texts. Ensure that data is encoded using UTF-8 during table creation and import.

14.2.29 Do I Need to Grant Table Permissions to a User and Project After I Delete a Table and Create One with the Same Name?

Scenario

User A created the **testTable** table in a database through a SQL job and granted user B the permission to insert and delete table data. User A deleted the **testTable** table and created a new **testTable** table. If user A wants user B to retain the insert and delete permission, user A needs to grant the permissions to user B again.

Possible Causes

After a table is deleted, the table permissions are not retained. You need to grant permissions to a user or project.

Solution

Operations to grant permissions to a user or project are as follows:

1. On the left of the management console, choose **Data Management** > **Databases and Tables**.
2. Click the database name whose table permission is to be granted. The table management page of the database is displayed.

3. Locate the row of the target table and click **Permissions** in the **Operation** column.
4. On the displayed page, click **Grant Permission** in the upper right corner.
5. In the displayed **Grant Permission** dialog box, select the required permissions.
6. Click **OK**.

14.2.30 Why Can't I Query Table Data After Data Is Imported to a DLI Partitioned Table Because the File to Be Imported Does Not Contain Data in the Partitioning Column?

Symptom

A CSV file is imported to a DLI partitioned table, but the imported file data does not contain the data in the partitioning column. The partitioning column needs to be specified for a partitioned table query. As a result, table data cannot be queried.

Possible Causes

When data is imported to a DLI partitionedtable, if the file data does not contain the partitioning column, the system specifies **__HIVE_DEFAULT_PARTITION__** as the column by default. If a Spark job finds that the partition is empty, **null** is returned.

Solution

1. Log in to the DLI management console. In the SQL editor, click **Settings**.
2. Add **spark.sql.forcePartitionPredicatesOnPartitionedTable.enabled** and set it to **false**.
3. Query the entire table or the partitioned table.

14.2.31 How Do I Fix the Data Error Caused by CRLF Characters in a Field of the OBS File Used to Create an External OBS Table?

Symptom

When an OBS foreign table is created, a field in the specified OBS file contains a carriage return line feed (CRLF) character. As a result, the data is incorrect.

The statement for creating an OBS foreign table is similar as follows:

```
CREATE TABLE test06 (name string, id int, no string) USING csv OPTIONS (path "obs://dli-test-001/test.csv");
```

The file contains the following information (example):

```
Jordon,88,"aa
bb"
```

A carriage return exists between **aa** and **bb** in the last field. As a result, he data in the **test06** table is displayed as follows:

```
name    id    classno
Jordon    88    aa
bb"    null    null
```

Solution

When creating an OBS foreign table, set **multiLine** to **true** to specify that the column data contains CRLF characters. The following is an example to solve the problem:

```
CREATE TABLE test06 (name string, id int, no string) USING csv OPTIONS (path "obs://dli-test-001/test.csv",multiLine=true);
```

14.2.32 Why Does a SQL Job That Has Join Operations Stay in the Running State?

Symptom

A SQL job contains join operations. After the job is submitted, it is stuck in the Running state and no result is returned.

Possible Causes

When a Spark SQL job has join operations on small tables, all executors are automatically broadcast to quickly complete the operations. However, this increases the memory consumption of the executors. If the executor memory usage is too high, the job fails to be executed.

Solution

1. Check whether the **/*+ BROADCAST(u) */** falg is used to forcibly perform broadcast join in the executed SQL statement. If the flag is used, remove it.
2. Set **spark.sql.autoBroadcastJoinThreshold** to **-1**.
   a. Log in to the DLI management console and choose **Job Management** > **SQL Jobs**. In the **Operation** column of the failed job, click **Edit** to switch to the SQL editor page.
   b. Click **Settings** in the upper right corner. In the **Parameter Settings** area, add **spark.sql.autoBroadcastJoinThreshold** and set it to **-1**.
   c. Click **Execute** again to and view the job running result.

14.2.33 The on Clause Is Not Added When Tables Are Joined. Cartesian Product Query Causes High Resource Usage of the Queue, and the Job Fails to Be Executed

Symptom

The on clause was not added to the SQL statement for joining tables. As a result, the Cartesian product query occurs due to multi-table association, and the queue resources were used up. Job execution fails on the queue.

For example, the following SQL statement left-joins three tables without the on clause.

```
select
    case
      when to_char(from_unixtime(fs.special_start_time), 'yyyy-mm-dd') < '2018-10-12' and row_number()
over(partition by fg.goods_no order by fs.special_start_time asc) = 1 then 1
      when to_char(from_unixtime(fs.special_start_time), 'yyyy-mm-dd') >= '2018-10-12' and fge.is_new = 1
then 1
      else 0 end as is_new
from testdb.table1 fg
left join testdb.table2 fs
left join testdb.table3  fge
where to_char(from_unixtime(fs.special_start_time), 'yyyymmdd') = substr('20220601',1,8)
```

Solution

When you use join to perform multi-table query, you must use the on clause to reduce the data volume.

The following example uses the on clause for the table join, which greatly reduces the result set of associated query and improves the query efficiency.

```
select
    case
      when to_char(from_unixtime(fs.special_start_time), 'yyyy-mm-dd') < '2018-10-12' and row_number()
over(partition by fg.goods_no order by fs.special_start_time asc) = 1 then 1
      when to_char(from_unixtime(fs.special_start_time), 'yyyy-mm-dd') >= '2018-10-12' and fge.is_new = 1
then 1
      else 0 end as is_new
from testdb.table1 fg
left join testdb.table2 fs on fg.col1 = fs.col2
left join testdb.table3  fge on fg.col3 = fge.col4
where to_char(from_unixtime(fs.special_start_time), 'yyyymmdd') = substr('20220601',1,8)
```

14.2.34 Why Can't I Query Data After I Manually Add Data to the Partition Directory of an OBS Table?

Symptom

Partition data is manually uploaded to a partition of an OBS table. However, the data cannot be queried using DLI SQL editor.

Solution

After manually adding partition data, you need to update the metadata information of the OBS table. Run the following statement on desired table:

```
MSCK REPAIR TABLE table_name;
```

Query the data in the OBS partitioned table.

14.2.35 Why Is All Data Overwritten When insert overwrite Is Used to Overwrite Partitioned Table?

To dynamically overwrite the specified partitioned data in the DataSource table, set **dli.sql.dynamicPartitionOverwrite.enabled** to **true** and then run the insert overwrite statement. (The default value of **dli.sql.dynamicPartitionOverwrite.enabled** is **false**.)

14.2.36 Why Is a SQL Job Stuck in the Submitting State?

The possible causes and solutions are as follows:

- After you purchase a DLI queue and submit a SQL job for the first time, wait for 5 to 10 minutes. After the cluster is started in the background, the submission will be successful.
- If the network segment of the queue is changed, wait for 5 to 10 minutes and then submit the SQL job immediately. After the cluster is re-created in the background, the submission is successful.
- The queue is idle for more than one hour, and background resources have been released. You must wait for 5 to 10 minutes and then submit the SQL job. After the cluster is restarted in the background, the submission will be successful.

14.2.37 Why Is the create_date Field in the RDS Table Is a Timestamp in the DLI query result?

Spark does not have the datetime type and uses the TIMESTAMP type instead.

You can use a function to convert data types.

The following is an example.

select cast(create_date as string), * from table where create_date>'2221-12-01 00:00:00';

14.2.38 What Can I Do If datasize Cannot Be Changed After the Table Name Is Changed in a Finished SQL Job?

If the table name is changed immediately after SQL statements are executed, the data size of the table may be incorrect.

If you need to change the table name, change it 5 minutes after the SQL job is complete.

14.2.39 Why Is the Data Volume Changes When Data Is Imported from DLI to OBS?

Symptom

When DLI is used to insert data into an OBS temporary table, only part of data is imported.

Possible Causes

Possible causes are as follows:

- The amount of data read during job execution is incorrect.
- The data volume is incorrectly verified.

Run a query statement to check whether the amount of imported data is correct.

If OBS limits the number of files to be stored, add **DISTRIBUTE BY number** to the end of the insert statement. For example, if **DISTRIBUTE BY 1** is added to the end of the insert statement, multiple files generated by multiple tasks can be inserted into one file.

Procedure

Step 1 On the DLI management console, check whether the number of results in the SQL job details is correct. The check result shows that the amount of data is correct.

Step 2 Check whether the method to verify the data volume is correct. Perform the following steps to verify the data amount:

1. Download the data file from OBS.
2. Use the text editor to open the data file. The data volume is less than the expected volume.

If you used this method, you can verify that the text editor cannot read all the data.

Run the query statement to view the amount of data import into the OBS bucket. The query result indicates that all the data is imported.

This issue is caused by incorrect verification of the data volume.

----End

14.3 Problems Related to Spark Jobs

14.3.1 Spark Jobs

Does DLI Spark Support Scheduled Periodic Jobs?

DLI Spark does not support job scheduling. You can use other services, such as DataArts Studio, or use APIs or SDKs to customize job schedule.

Can I Define the Primary Key When I Create a Table with a Spark SQL Statement?

The Spark SQL syntax does not support primary key definition.

Can DLI Spark Jar Jobs Access GaussDB(DWS) Datasource Tables?

Yes.

How Do I Check the Version of the Spark Built-in Dependency Package?

DLI built-in dependencies are provided by the platform by default. In case of conflicts, you do not need to upload them when packing JAR files of Spark or Flink Jar jobs.

Can I Download Packages on the Package Management Page?

No, the packages cannot be downloaded.

14.3.2 How Do I Use Spark to Write Data into a DLI Table?

To use Spark to write data into a DLI table, configure the following parameters:

- fs.obs.access.key
- fs.obs.secret.key
- fs.obs.impl
- fs.obs.endpoint

The following is an example:

```
import logging
from operator import add
from pyspark import SparkContext

logging.basicConfig(format='%(message)s', level=logging.INFO)

#import local file
test_file_name = "D://test-data_1.txt"
out_file_name = "D://test-data_result_1"

sc = SparkContext("local","wordcount app")
sc._jsc.hadoopConfiguration().set("fs.obs.access.key", "myak")
sc._jsc.hadoopConfiguration().set("fs.obs.secret.key", "mysk")
sc._jsc.hadoopConfiguration().set("fs.obs.impl", "org.apache.hadoop.fs.obs.OBSFileSystem")
sc._jsc.hadoopConfiguration().set("fs.obs.endpoint", "myendpoint")

# red: text_file rdd object
text_file = sc.textFile(test_file_name)

# counts
counts = text_file.flatMap(lambda line: line.split(" ")).map(lambda word: (word, 1)).reduceByKey(lambda a,
b: a + b)
# write
counts.saveAsTextFile(out_file_name)
```

14.3.3 How Do I Set the AK/SK for a Queue to Operate an OBS Table?

- If the AK and SK are obtained, set the parameters as follows:
  - Create SparkContext using code

    ```
    val sc: SparkContext = new SparkContext()
    sc.hadoopConfiguration.set("fs.obs.access.key", ak)
    sc.hadoopConfiguration.set("fs.obs.secret.key", sk)
    ```

  - Create SparkSession using code

    ```
    val sparkSession: SparkSession = SparkSession
        .builder()
        .config("spark.hadoop.fs.obs.access.key", ak)
        .config("spark.hadoop.fs.obs.secret.key", sk)
        .enableHiveSupport()
        .getOrCreate()
    ```

- If **ak**, **sk**, and **securitytoken** are obtained, the temporary AK/SK and security token must be used at the same time during authentication. The setting is as follows:
  - Create SparkContext using code

    ```
    val sc: SparkContext = new SparkContext()
    sc.hadoopConfiguration.set("fs.obs.access.key", ak)
    sc.hadoopConfiguration.set("fs.obs.secret.key", sk)
    sc.hadoopConfiguration.set("fs.obs.session.token", sts)
    ```

  - Create SparkSession using code

```
val sparkSession: SparkSession = SparkSession
    .builder()
    .config("spark.hadoop.fs.obs.access.key", ak)
    .config("spark.hadoop.fs.obs.secret.key", sk)
    .config("spark.hadoop.fs.obs.session.token", sts)
    .enableHiveSupport()
    .getOrCreate()
```

NOTE

For security purposes, you are advised not to include the AK and SK information in the OBS path. In addition, if a table is created in the OBS directory, the OBS path specified by the **Path** field cannot contain the AK and SK information.

14.3.4 How Do I View the Resource Usage of DLI Spark Jobs?

Viewing the Configuration of a Spark Job

Log in to the DLI console. In the navigation pane, choose **Job Management** > **Spark Jobs**. In the job list, locate the target job and click ⌄ next to Job ID to view the parameters of the job.

NOTE

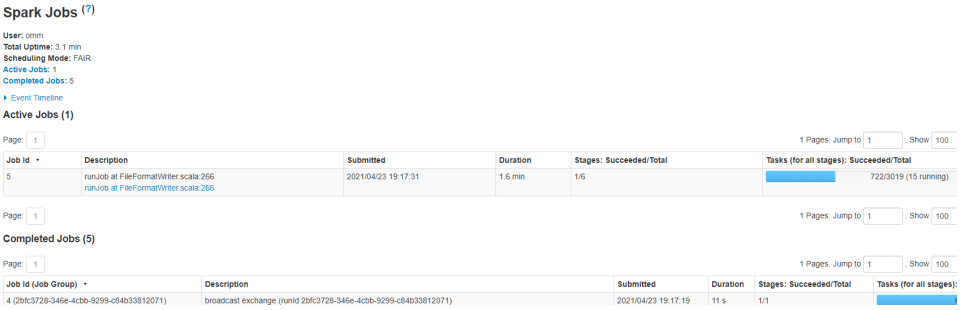
The content is displayed only when the parameters in **Advanced Settings** are configured during Spark job creation.

Viewing Real-Time Resource Usage of a Spark Job

Perform the following operations to view the number of running CUs occupied by a Spark job in real time:

1. Log in to the DLI console. In the navigation pane, choose **Job Management** > **Spark Jobs**. In the job list, locate the target job and click **SparkUI** in the **Operation** column.
2. On the Spark UI page, view the real-time running resources of the Spark job.

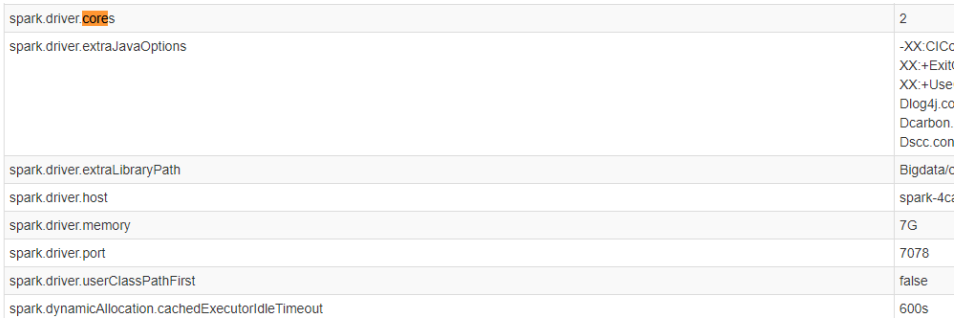
Figure 14-8 SparkUI



3. On the Spark UI page, view the original configuration of the Spark job (available only to new clusters).

   On the Spark UI page, click **Environment** to view **Driver** and **Executor** information.

Figure 14-9 Driver information



spark.driver.cores	2
spark.driver.extraJavaOptions	-XX:CICc XX:+ExitC XX:+Use Dlog4j.co Dcarbon. Dscc.con
spark.driver.extraLibraryPath	Bigdata/c
spark.driver.host	spark-4c
spark.driver.memory	7G
spark.driver.port	7078
spark.driver.userClassPathFirst	false
spark.dynamicAllocation.cachedExecutorIdleTimeout	600s

Figure 14-10 Executor information

spark.executor.cores	2
spark.executor.extraClassPath	hadoop/c
spark.executor.extraJavaOptions	-XX:CICo XX:+ExitC XX:+Use Dlog4j.co Dcarbon. Dscc.con
spark.executor.extraLibraryPath	Bigdata/c
spark.executor.heartbeatInterval	10000ms
spark.executor.id	driver
spark.executor.instances	7
spark.executor.memory	8G
spark.executor.memoryOverhead	2G
spark.executor.periodicGC.interval	30min

14.3.5 How Do I Use Python Scripts to Access the MySQL Database If the pymysql Module Is Missing from the Spark Job Results Stored in MySQL?

1. If the pymysql module is missing, check whether the corresponding EGG package exists. If the package does not exist, upload the pyFile package on the **Package Management** page. The procedure is as follows:
   a. Upload the **egg** package to the specified OBS path.
   b. Log in to the DLI management console and choose **Data Management** > **Package Management**.
   c. On the **Package Management** page, click **Create Package** in the upper right corner to create a package.
   d. In the **Create Package** dialog, set the following parameters:
      - Type: Select **PyFile**.
      - **OBS Path**: Select the OBS path where the **egg** package is stored.
      - Set **Group** and **Group Name** as you need.
   e. Click **OK**.
   f. On the Spark job editing page where the error is reported, choose the uploaded **egg** package from the **Python File Dependencies** drop-down list and run the Spark job again.
2. To interconnect PySpark jobs with MySQL, you need to create a datasource connection to enable the network between DLI and RDS.

For details about how to create a datasource connection on the management console, see "Enhanced Datasource Connections" in *Data Lake Insight User Guide*.

For details about how to call an API to create a datasource connection, see "Creating an Enhanced Datasource Connection" in *Data Lake Insight API Reference*.

14.3.6 How Do I Run a Complex PySpark Program in DLI?

DLI natively supports PySpark.

For most cases, Python is preferred for data analysis, and PySpark is the best choice for big data analysis. Generally, JVM programs are packed into JAR files and depend on third-party JAR files. Similarly, Python programs also depend on third-party libraries, especially big data analysis programs related to PySpark-based converged machine learning. Traditionally, the Python library is installed on the execution machine based on pip. For serverless services such as DLI, you do not need to and are unaware of the underlying compute resources. In this case, how does DLI ensure that you run their programs perfectly?

DLI has built-in algorithm libraries for machine learning in its compute resources. These common algorithm libraries meet the requirements of most users. What if a user's PySpark program depends on a program library that is not provided by the built-in algorithm library? Actually, the dependency of PySpark is specified based on PyFiles. On the DLI Spark job page, you can directly select the Python third-party program library (such as ZIP and EGG) stored on OBS.

The compressed package of the dependent third-party Python library has structure requirements. For example, if the PySpark program depends on moduleA (import moduleA), the compressed package must meet the following structure requirement:

Figure 14-11 Compressed package structure requirement

```
xxx.zip
  moduleA
    a.py
    b.py
    ...
```

That is, the compressed package contains a folder named after a module name, and then the Python file of the corresponding class. Generally, the downloaded Python library may not meet this requirement. Therefore, you need to compress the Python library again. In addition, there is no requirement on the name of the compressed package. Therefore, it is recommended that you compress the packages of multiple modules into a compressed package. Now, a large and complex PySpark program is configured and runs normally.

14.3.7 How Does a Spark Job Access a MySQL Database?

You can use DLI Spark jobs to access data in the MySQL database using either of the following methods:

- Solution 1: Buy a queue, create an enhanced datasource connection, and read data from the MySQL database through a datasource table. You need to write Java or Scala code to implement this solution.
- Solution 2: Use CDM to import data from the MySQL database to an OBS bucket, and then use a Spark job to read data from the OBS bucket. If you already have a CDM cluster, this solution is simpler than solution 1 and does not involve any other database.

14.3.8 How Do I Use JDBC to Set the spark.sql.shuffle.partitions Parameter to Improve the Task Concurrency?

Scenario

When shuffle statements, such as GROUP BY and JOIN, are executed in Spark jobs, data skew occurs, which slows down the job execution.

To solve this problem, you can configure **spark.sql.shuffle.partitions** to improve the concurrency of shuffle read tasks.

Configuring spark.sql.shuffle.partitions

You can use the **set** clause to configure the **dli.sql.shuffle.partitions** parameter in JDBC. The statement is as follows:

```
Statement st = conn.stamte()
st.execute("set spark.sql.shuffle.partitions=20")
```

14.3.9 How Do I Read Uploaded Files for a Spark Jar Job?

You can use SparkFiles to read the file submitted using **--file** form a local path: **SparkFiles.get(***"Name of the uploaded file"***)**.

NOTE

- The file path in the Driver is different from that obtained by the Executor. The path obtained by the Driver cannot be passed to the Executor.
- You still need to call **SparkFiles.get(***"filename"***)** in Executor to obtain the file path.
- The **SparkFiles.get()** method can be called only after Spark is initialized.

The java code is as follows:

```
package main.java

import org.apache.spark.SparkFiles
import org.apache.spark.sql.SparkSession

import scala.io.Source

object DliTest {
  def main(args:Array[String]): Unit = {
    val spark = SparkSession.builder
      .appName("SparkTest")
      .getOrCreate()

    // Driver: obtains the uploaded file.
    println(SparkFiles.get("test"))
```

```
  spark.sparkContext.parallelize(Array(1,2,3,4))
      // Executor: obtains the uploaded file.
    .map(_ => println(SparkFiles.get("test")))
    .map(_ => println(Source.fromFile(SparkFiles.get("test")).mkString)).collect()
 }
}
```

14.3.10 Why Are Errors "ResponseCode: 403" and "ResponseStatus: Forbidden" Reported When a Spark Job Accesses OBS Data?

Symptom

The following error is reported when a Spark job accesses OBS data:

```
Caused by: com.obs.services.exception.ObsException: Error message:Request Error.OBS servcie Error Message. -- ResponseCode: 403, ResponseStatus: Forbidden
```

Solution

Set the AK/SK to enable Spark jobs to access OBS data. For details, see **How Do I Set the AK/SK for a Queue to Operate an OBS Table?**

14.3.11 Why Is Error "verifyBucketExists on XXXX: status [403]" Reported When I Use a Spark Job to Access an OBS Bucket That I Have Access Permission?

Check whether the OBS bucket is used to store DLI logs on the **Global Configuration** > **Job Configurations** page. The job log bucket cannot be used for other purpose.

14.3.12 Why Is a Job Running Timeout Reported When a Spark Job Runs a Large Amount of Data?

When a Spark job accesses a large amount of data, for example, accessing data in a GaussDB(DWS) database, you are advised to set the number of concurrent tasks and enable multi-task processing.

14.3.13 Why Does the Job Fail to Be Executed and the Log Shows that the File Directory Is Abnormal When I Use a Spark Job to Access Files in SFTP?

Spark jobs cannot access SFTP. Upload the files you want to access to OBS and then you can analyze the data using Spark jobs.

14.3.14 Why Does the Job Fail to Be Executed Due to Insufficient Database and Table Permissions?

Symptom

When a Spark job is running, an error message is displayed, indicating that the user does not have the database permission. The error information is as follows:

```
org.apache.spark.sql.AnalysisException: org.apache.hadoop.hive.ql.metadata.HiveException:
MetaException(message:Permission denied for resource: databases.xxx,action:SPARK_APP_ACCESS_META)
```

Solution

You need to assign the database permission to the user who executes the job. The procedure is as follows:

1. In the navigation pane on the left of the management console, choose **Data Management** > **Databases and Tables**.
2. Locate the row where the target database resides and click **Permissions** in the **Operation** column.
3. On the displayed page, click **Grant Permission** in the upper right corner.
4. In the displayed dialog box, select **User** or **Project**, enter the username or select the project that needs the permission, and select the desired permissions.
5. Click **OK**.

14.3.15 Why Can't I Find the Specified Python Environment After Adding the Python Package?

I cannot find the specified Python environment after adding the Python 3 package.

Set **spark.yarn.appMasterEnv.PYSPARK_PYTHON** to **python3** in the **conf** file to specify the Python 3 environment for the compute cluster.

New clusters use the Python 3 environment by default.

14.3.16 Why Is a Spark Jar Job Stuck in the Submitting State?

The remaining CUs in the queue may be insufficient. As a result, the job cannot be submitted.

To view the remaining CUs of a queue, perform the following steps:

1. Check the CU usage of the queue.

   Log in to the Cloud Eye console. In the navigation pane on the left, choose **Cloud Service Monitoring** > **Data Lake Insight**. On the displayed page, locate the desired queue and click **View Metric** in the **Operation** column, and check **CU Usage (queue)** on the displayed page.

2. Calculate the number of remaining CUs.

   Remaining CUs of a queue = Total CUs of the queue – CU usage.

If the number of remaining CUs is less than the number of CUs required by the job, the job submission fails. The submission can be successful only after resources are available.

14.4 Product Consultation

14.4.1 What Is DLI?

Data Lake Insight (DLI) is a serverless data processing and analysis service fully compatible with Apache Spark and Apache Flink ecosystems. It frees you from managing any server. DLI supports standard SQL and is compatible with Spark and Flink SQL. It also supports multiple access modes, and is compatible with mainstream data formats. DLI supports SQL statements and Spark applications for heterogeneous data sources, including CloudTable, RDS, GaussDB(DWS), CSS, OBS, custom databases on ECSs, and offline databases.

14.4.2 Which Data Formats Does DLI Support?

DLI supports the following data formats:

- Parquet
- CSV
- ORC
- Json
- Avro

14.4.3 What Are the Differences Between MRS Spark and DLI Spark?

The Spark component of DLI is a fully managed service. You can only use the DLI Spark through its APIs. .

The Spark component of MRS is built on the VM in an MRS cluster. You can develop the Spark component to optimize it as needed and make API calls to use it.

14.4.4 Where Can DLI Data Be Stored?

DLI data can be stored in either of the following:

- OBS: Data used by SQL jobs, Spark jobs, and Flink jobs can be stored in OBS, reducing storage costs.
- DLI: The column-based **Parquet** format is used in DLI. That is, the data is stored in the **Parquet** format. The storage cost is relatively high.
- Datasource connection jobs can be stored in the connected services. Currently, CloudTable, CSS, DCS, DDS, GaussDB(DWS), MRS, and RDS are supported.

14.4.5 What Are the Differences Between DLI Tables and OBS Tables?

- DLI tables store data within the DLI service, and you do not need to know the data storage path.

- OBS tables store data in your OBS buckets, and you need to manage the source data files.
- Different from OBS tables, DLI tables give you more permission control and cache acceleration functions. The performance of DLI tables is better than that of foreign tables, but you will be charged for the storage.

14.4.6 How Can I Use DLI If Data Is Not Uploaded to OBS?

Currently, DLI supports analysis only on the data uploaded to the cloud. In scenarios where regular (for example, on a per day basis) one-off analysis on incremental data is conducted for business, you can do as follows: Anonymize data to be analyzed and store anonymized data on OBS temporarily. After analysis is complete, export the analysis report and delete the data temporarily stored on OBS.

14.4.7 Can I Import OBS Bucket Data Shared by Other Tenants into DLI?

Data in the OBS bucket shared by IAM users under the same account can be imported. You cannot import data in the OBS bucket shared with other IAM account.

14.4.8 Why Is Error "Failed to create the database. {"error_code":"DLI.1028";"error_msg":"Already reached the maximum quota of databases:XXX"." Reported?

Viewing My Quotas

1. Log in to the management console.
2. Click  in the upper left corner and select **Region** and **Project**.
3. Click  (the **My Quotas** icon) in the upper right corner.

   The **Service Quota** page is displayed.
4. View the used and total quota of each type of resources on the displayed page.

   If a quota cannot meet service requirements, increase a quota.

How Do I Apply for a Higher Quota?

The system does not support online quota adjustment. To increase a resource quota, dial the hotline or send an email to the customer service. We will process your application and inform you of the progress by phone call or email.

Before you contact customer service, prepare the following information:

- Account name, project name, and project ID

  Log in to the management console, click the username in the upper-right corner, choose **My Credentials**, and obtain the domain name, project name, and project ID.

- Quota information, including:
  - Service name
  - Quota type
  - Required quota

Learn how to obtain the service hotline and email address.

14.4.9 Can a Member Account Use Global Variables Created by Other Member Accounts?

No, a global variable can only be used by the user who created it. Global variables can be used to simplify complex parameters. For example, long and difficult variables can be replaced to improve the readability of SQL statements.

The restrictions on using global variables are as follows:

- Existing sensitive variables can only be used by their respective creators. Other common global variables are shared by users under the same account and project.
- If there are multiple global variables with the same name in the same project under an account, delete the redundant global variables to ensure that the global variables are unique in the same project. In this case, all users who have the permission to modify the global variables can change the variable values.
- If there are multiple global variables with the same name in the same project under an account, delete the global variables created by the user first. If there are only unique global variables, all users who have the delete permission can delete the global variables.

14.4.10 How Do I Manage Tens of Thousands of Jobs Running on DLI?

If you are suggested to perform following operations to run a large number of DLI jobs:

- Group the DLI jobs by type, and run each group on a queue.
- Alternatively, create IAM users to execute different types of jobs. .

14.4.11 How Do I Change the Name of a Field in a Created Table?

The field names of tables that have been created cannot be changed.

You can create a table, define new table fields, and migrate data from the old table to the new one.

14.4.12 Does DLI Have the Apache Spark Command Injection Vulnerability (CVE-2022-33891)?

No. The **spark.acls.enable** configuration item is not used in DLI. The Apache Spark command injection vulnerability (CVE-2022-33891) does not exist in DLI.

14.5 Quota

14.5.1 How Do I View My Quotas?

1. Log in to the management console.
2. Click  in the upper left corner and select **Region** and **Project**.
3. Click  (the **My Quotas** icon) in the upper right corner.
   The **Service Quota** page is displayed.
4. View the used and total quota of each type of resources on the displayed page.
   If a quota cannot meet service requirements, increase a quota.

14.5.2 How Do I Increase a Quota?

Increasing a Quota

The system does not support online quota adjustment. To increase a resource quota, dial the hotline or send an email to the customer service. We will process your application and inform you of the progress by phone call or email.

Before you contact customer service, prepare the following information:

- Account name, project name, and project ID
  Log in to the management console, click the username in the upper-right corner, choose **My Credentials**, and obtain the domain name, project name, and project ID.
- Quota information, including:
  - Service name
  - Quota type
  - Required quota

Learn how to obtain the service hotline and email address.

14.6 Permission

14.6.1 How Do I Manage Fine-Grained DLI Permissions?

DLI has a comprehensive permission control mechanism and supports fine-grained authentication through Identity and Access Management (IAM). You can create policies in IAM to manage DLI permissions.

With IAM, you can use your account to create IAM users for your employees, and assign permissions to the users to control their access to specific resource types. For example, some software developers in your enterprise need to use DLI resources but must not delete them or perform any high-risk operations. To

achieve this result, you can create IAM users for the software developers and grant them only the permissions required for using DLI resources.

NOTE

For a new user, you need to log in for the system to record the metadata before using DLI.

IAM can be used free of charge. You pay only for the resources in your account.

If the account has met your requirements, you do not need to create an independent IAM user for permission management. Then you can skip this section. This will not affect other functions of DLI.

DLI System Permissions

Table 14-1 lists all the system-defined roles and policies supported by DLI.

You can grant users permissions by using roles and policies.

- Roles: A type of coarse-grained authorization that defines permissions related to user responsibilities. This mechanism provides only a limited number of service-level roles for authorization. When using roles to grant permissions, you need to also assign other roles on which the permissions depend to take effect. Roles are not an ideal choice for fine-grained authorization and secure access control.
- Policies: A type of fine-grained authorization that defines permissions required to perform operations on specific cloud resources under certain conditions. This type of authorization is more flexible and ideal for secure access control. For example, you can grant DLI users only the permissions for managing a certain type of cloud servers.

Table 14-1 DLI system permissions

Role/Policy Name	Description	Category
DLI FullAccess	Full permissions for DLI.	System-defined policy
DLI ReadOnlyAccess	Read-only permissions for DLI. With read-only permissions, you can use DLI resources and perform operations that do not require fine-grained permissions. For example, create global variables, create packages and package groups, submit jobs to the default queue, create tables in the default database, create datasource connections, and delete datasource connections.	System-defined policy

Role/Policy Name	Description	Category
Tenant Administrator	Tenant administrator • Administer permissions for managing and accessing all cloud services. After a database or a queue is created, the user can use the ACL to assign rights to other users. • Scope: project-level service	System-defined role
DLI Service Admin	DLI administrator • Administer permissions for managing and accessing the queues and data of DLI. After a database or a queue is created, the user can use the ACL to assign rights to other users. • Scope: project-level service	System-defined role

14.6.2 What Is Column Permission Granting of a DLI Partition Table?

You cannot perform permission-related operations on the partition column of a partitioned table.

However, when you grant the permission of any non-partition column in a partitioned table to another user, the user gets the permission of the partition column by default.

When the user views the permission of the partition table, the permission of the partition column will not be displayed.

14.6.3 Why Does My Account Have Insufficient Permissions Due to Arrears?

When you submit a job, a message is displayed indicating that the job fails to be submitted due to insufficient permission caused by arrears. In this case, you need to check the roles in your token:

- **op_restrict**: The account permission is restricted due to insufficient balance. If your account balance is insufficient, the tokens of all online users under this account are revoked. If a user logs in to the system again, the **op_restrict** permission is added to the obtained token.
- **op_suspended**: Your account is suspended due to arrears or other reasons. If your account is in arrears, the tokens of all online users under this account are revoked. If a user logs in to the system again, the **op_suspended** permission is added to the obtained token, and user operations (excluding cloud service users) will be restricted.

If the two roles described about are in your token, user operations are restricted.

14.6.4 Why Does the System Display a Message Indicating Insufficient Permissions When I Update a Program Package?

Symptom

When the user update an existing program package, the following error information is displayed:

```
"error_code"*DLI.0003","error_msg":"Permission denied for resource 'resources. xxx', User = 'xxx', Action = "UPDATE_RESOURCE'."
```

Solution

You need to assign the package permission to the user who executes the job. The procedure is as follows:

1. In the left navigation pane of the DLI management console, choose **Data Management** > **Package Management**.
2. On the **Package Management** page, click **Manage Permission** in the **Operation** column of the package. The **User Permissions** page is displayed.
3. Click **Grant Permission** in the upper right corner of the page to authorize a user to access a package group or package. Select the **Update Group** permission.
4. Click **OK**.

14.6.5 Why Is Error "DLI.0003: Permission denied for resource..." Reported When I Run a SQL Statement?

Symptom

When the SQL query statement is executed, the system displays a message indicating that the user does not have the permission to query resources.

Error information: **DLI.0003: Permission denied for resource 'databases.dli_test.tables.test.columns.col1', User = '{UserName}', Action = 'SELECT'**

Solution

The user does not have the permission to query the table.

In the navigation pane on the left of the DLI console page, choose **Data Management** > **Databases and Tables**, search for the desired database table, view the permission configuration, and grant the table query permission to the user who requires it.

14.6.6 Why Can't I Query Table Data After I've Been Granted Table Permissions?

The table permission has been granted and verified. However, after a period of time, an error is reported indicating that the table query fails.

There are two possible reasons:

- The granted permission has been canceled.
- View the table creation time to check whether the table is deleted or re-created by others. If it has been deleted or re-created, the permission becomes invalid.

14.6.7 Will an Error Be Reported if the Inherited Permissions Are Regranted to a Table That Inherits Database Permissions?

If a table inherits database permissions, you do not need to regrant the inherited permissions to the table.

When you grant permissions on a table on the console:

- If you set **Authorization Object** to **User** and select the permissions that are the same as the inherited permissions, the system displays a message indicating that the permissions already exist and do not need to be regranted.
- If you set **Authorization Object** to **Project** and select the permissions that are the same as the inherited permissions, the system does not notify you of duplicate permissions.

14.6.8 Why Can't I Query a View After I'm Granted the Select Table Permission on the View?

Symptom

User A created Table1.

User B created View1 based on Table1.

After the **Select Table** permission on Table1 is granted to user C, user C fails to query View1.

Possible Causes

User B does not have the **Select Table** permission on Table1.

Solution

Grant the **Select Table** permission on Table1 to user B. Then, query View1 as user C again.

14.7 Queue

14.7.1 Does the Description of a DLI Queue Can Be Modified?

Currently, you are not allowed to modify the description of a created queue. You can add the description when purchasing the queue.

14.7.2 Will Table Data in My Database Be Lost If I Delete a Queue?

Deleting a queue does not cause table data loss in your database.

14.7.3 How Does DLI Ensure the Reliability of Spark Jobs When a Queue Is Abnormal?

You need to develop a mechanism to retry failed jobs. When a faulty queue is recovered, your application tries to submit the failed jobs to the queue again.

14.7.4 How Do I Monitor Queue Exceptions?

DLI allows you to subscribe to an SMN topic for failed jobs.

1. Log in to the DLI console.
2. In the navigation pane on the left, choose **Queue Management**.
3. On the **Queue Management** page, click **Create SMN Topic** in the upper left corner. .

14.7.5 How Do I View DLI Queue Load?

Scenario

To check the running status of the DLI queue and determine whether to run more jobs on that queue, you need to check the queue load.

Procedure

1. Search for Cloud Eye on the console.
2. In the navigation pane on the left, choose **Cloud Service Monitoring** > **Data Lake Insight**.
3. Select the queue you want to view.

14.7.6 How Do I Determine Whether There Are Too Many Jobs in the Current Queue?

Description

You need to check the large number of jobs in the **Submitting** and **Running** states on the queue.

Solution

Use Cloud Eye to view jobs in different states on the queue. The procedure is as follows:

1. Log in the management console and search for Cloud Eye.
2. In the navigation pane on the left, choose **Cloud Service Monitoring** > **Data Lake Insight**.

3. On the **Cloud Service Monitoring** page, click the queue name.
4. On the monitoring page, view the following metrics to check the job status:
   a. Number of jobs being submitted: Statistics of jobs in the **Submitting** state on the current queue
   b. Number of running jobs: Statistics of jobs in the **Running** state on the current queue
   c. Number of finished jobs: Statistics of jobs in the **Finished** state on the current queue

14.7.7 How Do I Switch an Earlier-Version Spark Queue to a General-Purpose Queue?

Currently, DLI provides two types of queues, **For SQL** and **For general use**. SQL queues are used to run SQL jobs. General-use queues are compatible with Spark queues of earlier versions and are used to run Spark and Flink jobs.

You can perform the following steps to convert an old Spark queue to a general purpose queue.

1. Purchase a general purpose queue again.
2. Migrate the jobs in the old Spark queue to the new general queue. That is, specify a new queue when submitting Spark jobs.
3. Release the old Spark queue, that is, delete it or unsubscribe it from the queue.

14.7.8 Why Cannot I View the Resource Running Status of DLI Queues on Cloud Eye?

DLI queues do not use resources or bandwidth when no job is running. In this case, the running status of DLI queues is not displayed on CES.

14.7.9 How Do I Allocate Queue Resources for Running Spark Jobs If I Have Purchased 64 CUs?

In DLI, 64 CU = 64 cores and 256 GB memory.

In a Spark job, if the driver occupies 4 cores and 16 GB memory, the executor can occupy 60 cores and 240 GB memory.

14.7.10 Why Is Error "Queue plans create failed. The plan xxx target cu is out of quota" Reported When I Schedule CU Changes?

Scenario

Queue plans create failed. The plan xxx target cu is out of quota is displayed when you create a scheduled scaling task.

Solution

The CU quota of the current account is insufficient. You need to apply for more quotas.

14.7.11 Why Is a Timeout Exception Reported When a DLI SQL Statement Fails to Be Executed on the Default Queue?

Symptom

After a SQL job was submitted to the default queue, the job runs abnormally. The job log reported that the execution timed out. The exception logs are as follows:

```
[ERROR] Execute DLI SQL failed. Please contact DLI service.
[ERROR] Error message:Execution Timeout
```

Possible Causes

The default queue is a public preset queue in the system for function trials. When multiple users submit jobs to this queue, traffic control might be triggered. As a result, the jobs fail to be submitted.

Solution

Buy a custom queue for your jobs. The procedure is as follows:

1. In the navigation pane of the DLI management console, choose **Queue Management**.
2. In the upper right corner of the **Queue Management** page, click **Buy Queue** to create a queue.
3. On the **Buy Queue** page, set the required parameters as you need. Especially, set **Type** to **For SQL**.
4. Submit your SQL jobs to the newly created queue.

14.8 Datasource Connections

14.8.1 Why Do I Need to Create a VPC Peering Connection for an Enhanced Datasource Connection?

You need to create a VPC peering connection to enable network connectivity. Take MRS as an example. If DLI and MRS clusters are in the same VPC, and the security group is enabled, you do not need a VPC peering connection for communications between MRS and DLI.

14.8.2 Failed to Bind a Queue to an Enhanced Datasource Connection

Symptom

An enhanced datasource connection failed to pass the network connectivity test. Datasource connection cannot be bound to a queue. The following error information is displayed:

```
Failed to get subnet 86ddcf50-233a-449d-9811-cfef2f603213. Response code : 404, message : {"code":"VPC.0202","message":"Query resource by id 86ddcf50-233a-449d-9811-cfef2f603213 fail.the subnet could not be found."}
```

Cause Analysis

VPC Administrator permissions are required to use the VPC, subnet, route, VPC peering connection, and port for DLI datasource connections.

The binding fails because the user does not have the required VPC permissions.

Procedure

On the DLI console, choose **Global Configuration** > **Service Authorization**, select the required VPC permission, and click **Update**.

14.8.3 DLI Failed to Connect to GaussDB(DWS) Through an Enhanced Datasource Connection

Symptom

The outbound rule had been configured for the security group of the queue associated with the enhanced datasource connection. The datasource authentication used a password. The connection failed and **DLI.0999: PSQLException: The connection attempt failed** is reported.

Cause Analysis

Possible causes are as follows:

- The security group configuration is incorrect.
- The subnet configuration is incorrect.

Procedure

Step 1 Check whether the security group is accessible.

- Inbound rule: Check whether the inbound CIDR block and port in the security group have been enabled. If not, create the CIDR block and port you need.
- Outbound rule: Check whether the CIDR block and port of the outbound rule are enabled. (It is recommended that all CIDR blocks be enabled.)

Both the inbound and outbound rules of the security group are configured for the subnets of the DLI queue. Set the source IP address in the inbound direction to 0.0.0.0/0 and port 8000, indicating that any IP address can access port 8000.

Step 2 If the fault persists, check the subnet configuration. Check the network ACL associated with the GaussDB(DWS) subnet. A network ACL is an optional layer of security for your subnets. You can associate one or more subnets with a network ACL to control traffic in and out of the subnets. After the association, the network ACL denies all traffic to and from the subnet by default until you add rules to allow traffic. The check result showed that the ACL associated with the subnet where GaussDB(DWS) resides is empty.

A network ACL is associated and no inbound or outbound rules are configured. As a result, the IP address cannot be accessed.

Step 3 Perform the connectivity test. After the subnet inbound and outbound rules are configured, the datasource connection passes the connectivity test.

----End

14.8.4 How Do I Do if the Datasource Connection Is Created But the Network Connectivity Test Fails?

Description

A datasource connection is created and bound to a queue. The connectivity test fails and the following error information is displayed:

```
failed to connect to specified address
```

Fault Locating

The issues here are described in order of how likely they are to occur.

Troubleshoot the issue by ruling out the causes described here, one by one.

- **Check Whether a Port Number Is Added to the End of the Domain Name or IP Address**
- **Check Whether the Information of the Peer VPC and Subnet Are Correct.**
- **Check Whether the CIDR Block of the Queue Overlaps That of the Data Source**
- **Check Whether the VPC Administrator Permission Is Granted to DLI**
- **Check Whether the Destination Security Group Allows Access from the CIDR Block of the Queue**
- **Check the Route Information of the VPC Peering Connection Corresponding to an Enhanced Datasource Connection**
- **Check Whether VPC Network ACL Rules Are Configured to Restrict Network Access**

Check Whether a Port Number Is Added to the End of the Domain Name or IP Address

The port number is required for the connectivity test.

The following example tests the connectivity between a queue and a specified RDS DB instance. The RDS DB instance uses port 3306.

The following figure shows how you should specify the IP address.

Check Whether the Information of the Peer VPC and Subnet Are Correct.

When you create an enhanced datasource connection, you need to specify the peer VPC and subnet.

For example, to test the connectivity between a queue and a specified RDS DB instance, you need to specify the RDS VPC and subnet information.

Check Whether the CIDR Block of the Queue Overlaps That of the Data Source

The CIDR block of the DLI queue bound with a datasource connection cannot overlap the CIDR block of the data source.

You can check whether they overlap by viewing the connection logs.

CIDR block conflicts of queue A and queue B. In this example, queue B is bound to an enhanced datasource connection to data source C. Therefore, a message is displayed, indicating that the network segment of queue A conflicts with that of data source C. As a result, a new enhanced datasource connection cannot be established.

Solution: Modify the CIDR block of the queue or create another queue.

Planing the CIDR blocks for your queues helps you to avoid this problem.

Check Whether the VPC Administrator Permission Is Granted to DLI

View the connection logs to check whether there is the required permission.

Figure 14-12 and **Figure 14-13** show the logs when subnet ID and route ID of the destination cannot be obtained because there is no permission.

Solution: Grant DLI the VPC Administrator permission and cancel the IAM ReadOnlyAccess authorization.

Figure 14-12 Viewing connection logs

Figure 14-13 Viewing connection logs

Check Whether the Destination Security Group Allows Access from the CIDR Block of the Queue

To connect to Kafka, GaussDB(DWS), and RDS instances, add security group rules for the DLI CIDR block to the security group where the instances belong. For example, to connect a queue to RDS, perform the following operations:

1. Log in to the DLI console, choose **Resources** > **Queue Management** in the navigation pane on the left. On the displayed page, select the target queue, and click  to expand the row containing the target queue to view its CIDR block.
2. On the **Instance Management** page of the RDS console, click the instance name. In the **Connection Information** area, locate **Database Port** to obtain the port number of the RDS DB instance.
3. In the **Connection Information** area locate the **Security Group** and click the group name to switch to the security group management page. Select the **Inbound Rules** tab and click **Add Rule**. Set the priority to 1, protocol to TCP, port to the database port number, and source to the CIDR block of the DLI queue. Click **OK**.

Check the Route Information of the VPC Peering Connection Corresponding to an Enhanced Datasource Connection

Check the routing table of the VPC peering connection corresponding to the enhanced datasource connection. Check whether the CIDR block of the queue overlaps other CIDR blocks in the routing table. If it does, the forwarding may be incorrect.

1. Obtain the ID of the VPC peering connection created for the enhanced datasource connection.
2. View the information about the VPC peering connection on the VPC console.
3. View the route table information of the VPC corresponding to the queue.

Check Whether VPC Network ACL Rules Are Configured to Restrict Network Access

Check whether an ACL is configured for the subnet corresponding to the datasource connection and whether the ACL rules restrict network access.

For example, if you set a CIDR block whose security group rule allows access from a queue and set a network ACL rule to deny access from that CIDR block, the security group rule does not take effect.

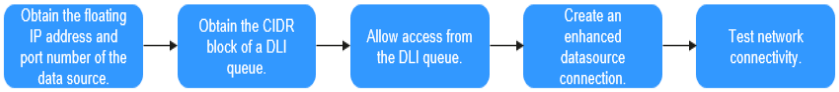
14.8.5 How Do I Configure the Network Between a DLI Queue and a Data Source?

- **Configuring the Connection Between a DLI Queue and a Data Source in a Private Network**

  If your DLI job needs to connect to a data source, for example, MRS, RDS, CSS, Kafka, or GaussDB(DWS), you need to enable the network between DLI and the data source.

An enhanced datasource connection uses VPC peering to directly connect the VPC networks of the desired data sources for point-to-point data exchanges.

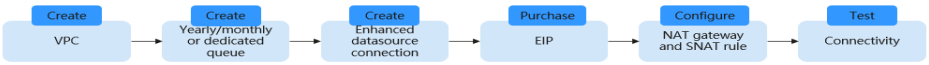
Figure 14-14 Configuration process



- **Configuring the Connection Between a DLI Queue and a Data Source in the Internet**

  You can configure SNAT rules and add routes to the public network to enable communications between a queue and the Internet.

Figure 14-15 Configuration process



14.8.6 What Can I Do If a Datasource Connection Is Stuck in Creating State When I Try to Bind a Queue to It?

The possible causes and solutions are as follows:

- If you have created a queue, do not bind it to a datasource connection immediately. Wait for 5 to 10 minutes. After the cluster is started in the background, the queue can be bound to the datasource connection.
- If you have changed the network segment of a queue, do not bind it to a datasource connection immediately. Wait for 5 to 10 minutes. After the cluster is re-created in the background, the creation is successful.

14.8.7 How Do I Connect DLI to Data Sources?

DLI enhanced datasource connection uses VPC peering to directly connect the VPC networks of the desired data sources for point-to-point data exchanges.

14.8.8 How Can I Perform Query on Data Stored on Services Rather Than DLI?

To perform query on data stored on services rather than DLI, perform the following steps:

1. Assume that the data to be queried is stored on multiple services (for example, OBS) on cloud.
2. On the DLI management console, create a table and set the **Path** parameter to the save path of the target data, for example, path of an OBS bucket. The data is actually stored on OBS and data migration is not required.
3. On the DLI management console, edit SQL statements to query and analyze the data.

14.8.9 How Can I Access Data Across Regions?

1. Connect VPCs in different regions.
2. Create an enhanced datasource connection on DLI and bind it to a queue.
3. Add a DLI route.

14.8.10 How Do I Set the Auto-increment Primary Key or Other Fields That Are Automatically Filled in the RDS Table When Creating a DLI and Associating It with the RDS Table?

When data is inserted into DLI, set the **ID** field to **NULL**.

14.8.11 Why Is the Error Message "communication link failure" Displayed When I Use a Newly Activated Datasource Connection?

- Possible Causes

  The network connectivity is abnormal. Check whether the security group is correctly selected and whether the VPC is correctly configured.

- Solution

  Example: When you create an RDS datasource connection, the system displays the error message **Communication link failure**.

  a. Delete the original datasource connection and create a new one. When you create a new connection, ensure that the selected **Security Group**, **VPC**, **Subnet**, and **Destination Address** are the same as those in RDS.

  NOTE

  Select a correct **Service Type**. In this example, select **RDS**.

  b. Check the configurations of VPC.

  If the error message is still displayed after you create a new datasource connection according to **Step 1**, check the VPC configuration.

14.8.12 Connection Times Out During MRS HBase Datasource Connection, and No Error Is Recorded in Logs

The cluster host information is not added to the datasource connection. As a result, the KRB authentication fails, the connection times out, and no error is recorded in logs. Configure the host information and try again.

On the **Enhanced** page, select the connection and click **Modify Host**. In the dialog box that is displayed, enter the host information. The format is **Host IP address Host name/Domain name**. Multiple records are separated by line breaks.

For details, see "Modifying the Host Information" in *Data Lake Insight User Guide*.

14.8.13 Why Can't I Find the Subnet When Creating a DLI Datasource Connection?

When you create a VPC peering connection for the datasource connection, the following error information is displayed:

```
Failed to get subnet 2c2bd2ed-7296-4c64-9b60-ca25b5eee8fe. Response code : 404, message : {"code":"VPC.0202","message":"Query resource by id 2c2bd2ed-7296-4c64-9b60-ca25b5eee8fe fail.the subnet could not be found."}
```

Before you create a datasource connection, check whether **VPC Administrator** is selected. If only the global **Tenant Administrator** is selected, the system cannot find the subnet.

14.8.14 Error Message "Incorrect string value" Is Displayed When insert overwrite Is Executed on a Datasource RDS Table

Symptom

A datasource RDS table was created in the DataArts Studio, and the **insert overwrite** statement was executed to write data into RDS. **DLI.0999: BatchUpdateException: Incorrect string value: '\xF0\x9F\x90\xB3' for column 'robot_name' at row 1** was reported.

Cause Analysis

The data to be written contains emojis, which are encoded in the unit of four bytes. MySQL databases use the UTF-8 format, which encodes data in the unit of three bytes by default. In this case, an error occurs when the emoji data is inserted into to the MySQL database.

Possible causes are as follows:

- A database coding error occurred.

Procedure

Change the character set to **utf8mb4**.

Step 1 Run the following SQL statement to change the database character set:

```
ALTER DATABASE DATABASE_NAME DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
```

Step 2 Run the following SQL statement to change the table character set:

```
ALTER TABLE TABLE_NAME DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
```

Step 3 Run the following SQL statement to change the character set for all fields in the table:

```
ALTER TABLE TABLE_NAME CONVERT TO CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
```

----End

14.8.15 Null Pointer Error Is Displayed When the System Creates a Datasource RDS Table

Symptom

The system failed to create a datasource RDS table, and null pointer error was reported.

Cause Analysis

The following table creation statement was used:

```
CREATE TABLE IF NOT EXISTS dli_to_rds
 USING JDBC OPTIONS (
 'url'='jdbc:mysql://to-rds-1174405119-oLRHAGE7.datasource.com:5432/postgreDB',
 'driver'='org.postgresql.Driver',
 'dbtable'='pg_schema.test1',
 'passwdauth' = 'xxx',
 'encryption' = 'true');
```

The RDS database is in a PostGre cluster, and the protocol header in the URL is invalid.

Procedure

Change the URL to **url'='jdbc:postgresql://to-rds-1174405119-oLRHAGE7.datasource.com:5432/postgreDB** and run the creation statement. The datasource table is successfully created.

14.8.16 Error Message "org.postgresql.util.PSQLException: ERROR: tuple concurrently updated" Is Displayed When the System Executes insert overwrite on a Datasource GaussDB(DWS) Table

Symptom

The system failed to execute **insert overwrite** on the datasource GaussDB(DWS) table, and **org.postgresql.util.PSQLException: ERROR: tuple concurrently updated** was displayed.

Cause Analysis

Concurrent operations existed in the job. Two insert overwrite operations were executed on the table at the same time.

One CN was running the following statement:

```
TRUNCATE TABLE BI_MONITOR.SAA_OUTBOUND_ORDER_CUST_SUM
```

Another CN was running the following command:

```
call bi_monitor.pkg_saa_out_bound_monitor_p_saa_outbound_order_cust_sum
```

This function deletes and inserts SAA_OUTBOUND_ORDER_CUST_SUM.

Procedure

Modify job logic to prevent concurrent insert overwrite operations on the same table.

14.8.17 RegionTooBusyException Is Reported When Data Is Imported to a CloudTable HBase Table Through a Datasource Table

Symptom

A datasource table was used to import data to a CloudTable HBase table. This HBase table contains a column family and a rowkey for 100 million simulating data records. The data volume is 9.76 GB. The job failed after 10 million data records were imported.

Cause Analysis

1. View driver error logs.
2. View executor error logs.
3. View task error logs.

The rowkey was poorly designed causing a large amount of traffic redirected to single or very few numbers of nodes.

Procedure

1. Pre-partition the HBase.
2. Hash the rowkey.

Summary and Suggestions

Distribute data to different RegionServer. Add **distribute by rand()** to the end of the insert statement.

14.8.18 A Null Value Is Written Into a Non-Null Field When a DLI Datasource Connection Is Used to Connect to a GaussDB(DWS) Table

Symptom

A table was created on GaussDB(DWS) and then a datasource connection was created on DLI to read and write data. An error message was displayed during data writing, indicating that DLI was writing a null value to a non-null field of the table, and the job failed.

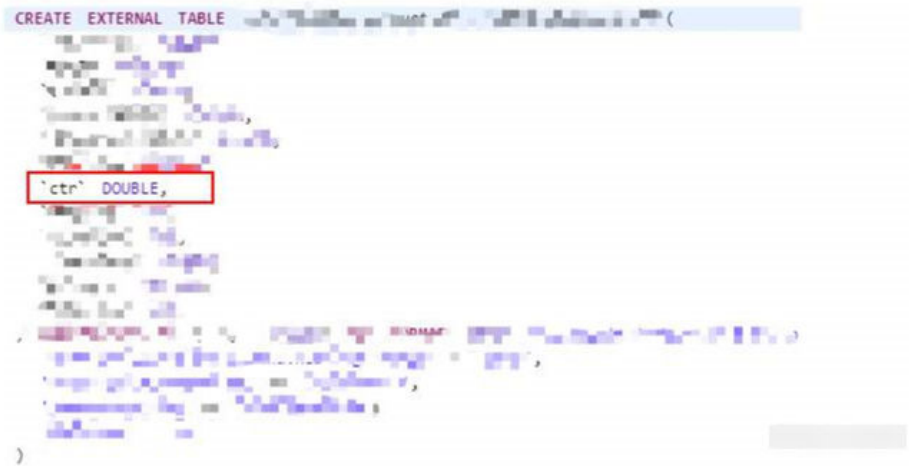
The error message is as follows:

```
DLI.0999: PSQLException: ERROR: dn_6009_6010: null value in column "ctr" violates not-null constraint
Detail: Failing row contains (400070309, 9.00, 25, null, 2020-09-22, 2020-09-23 04:30:01.741).
```

Cause Analysis

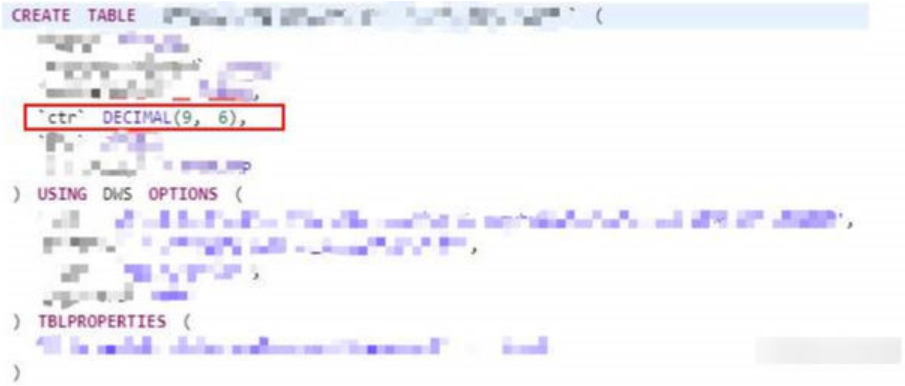
1. The CIR field in the source table is of the **DOUBLE** type.

   Figure 14-16 Creation statement of the source table

2. The field type in the target table is **DECIMAL(9,6)**.

   Figure 14-17 Creation statement of the target table

3. View the source table data. The CTR value that causes the problem is **1675**, which exceed the precision (9 – 6 = 3 digits) of the **DECIMAL(9,6)** type. A null value was generated when the double value was converted to the decimal value, and the insert operation failed.

Procedure

Change the precision of the decimal data defined in the target table.

14.8.19 An Insert Operation Failed After the Schema of the GaussDB(DWS) Source Table Is Updated

Symptom

A datasource GaussDB(DWS) table and the datasource connection were created in DLI, and the schema of the source table in GaussDB(DWS) were updated. During

the job execution, the schema of the source table failed to be updated, and the job failed.

Cause Analysis

When the insert operation is executed on the DLI datasource table, the GaussDB(DWS) source table is deleted and recreated. If the statement for creating the datasource table is not updated on DLI, the GaussDB(DWS) source table will fail to be updated.

Procedure

Create a datasource table on DLI and add table creation configuration **truncate = true** to clear table data but not delete the table.

Summary and Suggestions

After the source table is updated, the corresponding datasource table must be updated too on DLI.

14.9 APIs

14.9.1 Why Is Error "unsupported media Type" Reported When I Subimt a SQL Job?

In the REST API provided by DLI, the request header can be added to the request URI, for example, **Content-Type**.

Content-Type indicates the request body type or format. The default value is **application/json**.

URI for submitting a SQL job: **POST /v1.0/{project_id}/jobs/submit-job**

Content-Type can be only **application/json**. If **Content-Type** is set to **text**, "unsupported media Type" is displayed.

14.9.2 Are Project IDs of Different Accounts the Same When They Are Used to Call APIs?

If different IAM accounts call APIs in the same enterprise project in the same region, the accounts use the same project ID.

14.9.3 What Can I Do If an Error Is Reported When the Execution of the API for Creating a SQL Job Times Out?

Symptom

When the API call for submitting a SQL job times out, and the following error information is displayed:

```
There are currently no resources tracked in the state, so there is nothing to refresh.
```

Possible Causes

The timeout of API calls in synchronous is two minutes. If a call times out, an error will be reported.

Solution

When you make a call of the API for submitting a SQL job, set **dli.sql.sqlasync.enabled** to **true** to run the job asynchronously.

14.10 SDKs

14.10.1 How Do I Set the Timeout Duration for Querying SQL Job Results Using SDK?

When you query the SQL job results using SDK, the system checks the job status when the job is submitted. The timeout interval set in the system is 300s. If the job is not in the **FINISHED** state, a timeout error is reported after 300s.

You are advised to use **getJobId()** to obtain the job ID and then call **queryJobResultInfo(String jobId)** or **cancelJob(String jobId)** to obtain the result or cancel the job.

14.10.2 How Do I Handle the dli.xxx,unable to resolve host address Error?

1. Run the ping command to check whether **dli.xxx** can be accessed.

   If **dli.xxx** can be accessed, check whether DNS resolution is correctly configured.

2. DLI does not support cross-region access.

A Change History

Released On	What's New
2023-10-24	This issue is the third official release. Updated the following content: • Added restrictions on DLI-related functions to **Constraints**. • **Enhanced Datasource Connections** • **Datasource Authentication**
2023-05-22	This issue is the second official release. Added **FAQ**.
2021-08-15	This issue is the first official release.